

Small or medium-scale focused research project (STREP)



ICT Call 8

FP7-ICT-2011-8

**Cooperative Self-Organizing System for low Carbon Mobility at
low Penetration Rates**

The logo for the COLOMBO project. The word 'COLOMBO' is written in large, bold, black capital letters. The 'O's are replaced by circular icons: the first 'O' is red with a black grid pattern, the second 'O' is orange with a black grid pattern, and the third 'O' is green with a black grid pattern. The background of the logo features a green and yellow gradient with concentric white circles.

COLOMBO: Deliverable 1.2

**Data Collection and Dissemination for Low Penetration
Systems**

Document Information	
Title	Deliverable 1.2 - Data Collection and Dissemination for Low Penetration Systems
Dissemination Level	PU (Public)
Version	1.1
Date	17.11.2014
Status	Final version

Authors	Jérôme Härrı (EURECOM), Thrasyvoulos Spyropoulos (EURECOM), Luca Foschini (UniBo), Paolo Bellavista (UniBo), Daniel Krajzewicz (DLR), Robbin Blokpoel (IMTECH), Wolfgang Niebel (DLR) , Laura Bieker (DLR), Peter Wagner (DLR)
---------	--

Table of contents

1	Introduction.....	5
1.1	Document Objectives	6
1.2	Document Structure.....	6
2	State of the Art.....	7
2.1	Traffic Surveillance	7
2.1.1	Conventional, Static Sensors.....	7
2.1.2	Cooperative Traffic Surveillance	8
2.1.3	Clustering and Topology Management.....	10
2.2	Communication & Networking	10
2.2.1	Frequency Allocations	10
2.2.2	Investigated Technologies.....	12
2.3	V2X-based Traffic Surveillance Approaches.....	16
2.3.1	V2X communication-based.....	16
2.3.2	FP7 iTETRIS.....	18
3	Scenarios and Use Cases	22
3.1	Traffic Scenarios	22
3.1.1	Synthetic Intersection Scenario.....	22
3.1.2	Bologna Urban Scenarios.....	22
3.2	Traffic Light Scenarios.....	24
3.2.1	Self-Organizing Traffic-Light Control System.....	24
3.2.2	Virtual Sensors	25
3.2.3	Data Quality Requirements for Traffic Monitoring.....	26
4	V2X-based Traffic State Estimation.....	28
4.1	Algorithm 1: Cluster-based Traffic Surveillance	28
4.1.1	Reactive Group Formation Protocol	29
4.1.2	Proactive Group Formation Protocol	30
4.1.3	Group Lifecycle Management	31
4.1.4	Implementation in iCS	32
4.1.5	Simulation Results	33
4.2	Algorithm 2: DissFlow Traffic Surveillance.....	35
4.2.1	System Description	35
4.2.2	Dissemination Protocol	36
4.2.3	Dissemination Delay-Density Mapping Function.....	38
4.2.4	Implementation Details	40
4.2.5	Simulation Results	41
4.2.6	Summary and Outlook	42

4.3	Algorithm 3: Traffic State Generation from CAMs	42
4.3.1	System Description	43
4.3.2	Mapping Vehicles on Roads	44
4.3.3	Performance Indicators Computation	44
4.3.4	Simulation Results	45
4.3.5	Summary and Outlook	49
4.4	Algorithm 4: Probe Vehicle Data-based Traffic Surveillance	50
5	Outlook at Further Surveillance Applications	53
5.1	Driving Anomalies and Hazard Detection	53
5.2	Local Emissions Monitoring	54
6	Cost Analysis	56
7	Summary	58
	Appendix A – References	60
	Appendix B – Further iTETRIS Platform Extensions	64
B.1	Application Module Extensions	64
B.2	iCS optimization.....	65

1 Introduction

The COLOMBO project will deliver a set of modern cooperative traffic surveillance and control applications that target at different transport related objectives, such as increasing mobility, resource efficiency, and environmental friendliness.

Traffic management relies on knowledge about the state on the road, assembled from data and information collected by traffic monitoring and surveillance. Thereby these processes are the first step in the traffic management chain as can be seen exemplary in [Hoogendorn et al., 2011] or the German ITS Action Plan [BMVBS, 2012] (page 8). Conventionally, traffic surveillance is performed by road authorities and is used for both, strategic decisions as well as for triggering actions on tactical level. COLOMBO does not target to support strategic decisions, albeit the results from the project may be used for such applications as well. On tactical level, traffic management uses information about the current state of the road to a) monitor traffic state for recognizing incidents like flow breakdowns, b) automatically adapt traffic lights to the current demand, and c) assist in navigation. Tactical level traffic surveillance is envisioned to be critical to reach smart and sustainable mobility in future smart cities. It is expected to reduce the danger on the road, the commuting time as well as the traffic pollution, fuel consumption or even road wear and tear. Tackling these objectives is yet challenging, as traffic demand is assumed to increase, while the budgets of public authorities decrease. It is therefore meaningful to develop methods for traffic management that would at the same time be capable to monitor the increasing traffic demand and to reduce the costs.

To this objective, traffic engineering integrated Information Technology (IT) solutions and proposed a cloud-based traffic surveillance system called “Floating Car Data” in 2007. Vehicles that belong to an FCD fleet send their data, usually containing their positions, to a processing centre. The obtained positions are mapped on a digital road network to obtain travel times for passed roads. Irrespective if data is based on cellular or GPS location, FCD are very efficient for strategic decisions, such as macroscopic traffic monitoring and/or alternative routing. Tactical-level decisions require a reactivity that may not be provided by FCD due to a low penetration rate and long update intervals.

COLOMBO develops methods for tactical traffic surveillance, which allow gathering information about the current state on the roads at low costs. This is mainly achieved by exploiting data that is available from vehicular communication (V2X) that is aimed to be introduced in 2015.

COLOMBO therefore develops traffic surveillance systems falling in a new category called ‘*Distributed Floating Car Data*’ (DFCD). In DFCD, vehicles and smartphones are sampling traffic data, but instead of transferring these data to a central processing server, they use V2X technology to autonomously compute local traffic information and transmit them directly to traffic lights. The main objective of the local computation is not to reduce traffic on the communication and cloud systems, but to be able to work without it.

Different scenarios for V2X deployment exist, but within the first years of availability, only few equipped vehicles will be found in real-world networks. The algorithms developed in COLOMBO take this into account assuming a low penetration rate of equipped vehicles only. To nonetheless achieve valid information about the state on the roads, other wireless communication devices are incorporated, mainly Smart Phones or PDAs.

COLOMBO’s DFCD will have several innovative features compared to FCD:

- **Local Data Processing** – a major discouraging aspect of data crowdsourcing come from the lack of control of data, once it reaches the cloud. With COLOMBO DFCD, data remains at vehicles and at traffic lights.

- **Fresh Data Processing** – traffic flows are immediately and continuously monitored. The DFCD short delay and high reactivity is reached through local processing and direct transmission to traffic lights.
- **Low Cost** – FCD typically imply a data transfer (e.g. via SMS) flat rate contract. Unlike FCD, DFCD does not need vehicles either being in coverage of a cellular network, or have a roaming contract with them.
- **High penetration** – FCD are application dependent; TomTom IQ, Google Maps or any other app requires vehicles to voluntarily enrol and support the data cost, which can make sampling segmentation between app providers. COLOMBO's DFCD not depending on communication fees, enrolling to COLOMBO DFCD would reach all vehicles approaching intersections.
- **Adaptability** – COLOMBO's DFCD does not depend on vehicles being under coverage or willing to pay. The COLOMBO system only needs to place a TLC supporting dedicated vehicular communication to obtain traffic samples.
- **Multi-granularity** – V2X equipped vehicles will not provide a sufficient coverage and will be complemented by other dedicated technologies, such as WLAN and Bluetooth available in smartphones and/or vehicles.

On top of these solutions, COLOMBO aims on developing local incident and local emission monitoring systems. They will be presented in the subsequent deliverable D1.3 of this work package.

1.1 Document Objectives

The major objective of this deliverable is to present the traffic surveillance solutions developed in COLOMBO's Work Package (WP) 1. To put this work into the context of existing solutions and work, a summary on available traffic surveillance technologies is given, focussing on those that build upon vehicular communication. As well, this document shall give an outlook on the subsequent steps of this WP, mainly attempts to deliver a system for driving anomalies detection and a system for local emissions monitoring.

1.2 Document Structure

In Chapter 2, the currently available conventional traffic surveillance methods are described as well as traffic surveillance approaches based on vehicular communications. Afterwards, the scenarios used to evaluate the developed algorithms' performance are given in Chapter 3. Chapter 4 presents the traffic surveillance solutions developed in COLOMBO. In Chapter 5, an outlook on possibilities to detect incidents is given and the developed local emissions monitoring system is outlined. Chapter 6 gives an estimation of the costs of the developed solutions comparing them to conventional sensing technology. Chapter 7 summarizes the described work.

2 State of the Art

The following sections describe the currently used traffic surveillance technology, as well as the V2X-technology used by the solutions developed in COLOMBO.

2.1 Traffic Surveillance

Traffic surveillance is conventionally based on stationary road-side detectors. In urban areas, most detectors give their information directly to a traffic light. Therefore, they are usually close to specific traffic lights. Being part of the traffic light system, they are operated by cities. Cellphones belonging to drivers offered mobile phone operators the opportunity to enter this market. With their 'Floating Car Data', they rely on cellular triangulation to track vehicles and to monitor traffic. More recently, the generalization of GPS-enabled smartphones and cloud-based services also let data mining companies propose traffic surveillance solutions. This section will provide an overview of the available technologies and the evolution of the market.

2.1.1 Conventional, Static Sensors

The following summary on conventional, stationary sensors is widely based on the book "Traffic Simulation and Data: Validation Methods and Applications" [Daamen, Buisson, Hoogendoorn, 2014] (Chapter 2), which is one of the results of the COST-action "Multitude". The following technologies are used:

- inductive loop detectors,
- magnetic field detectors,
- pressure detectors,
- weigh-in-motion systems and piezoelectric sensors,
- radar detectors (Doppler and presence)
- passive and active infrared sensors,
- passive and active acoustic sensors,
- as well as camera-based virtual loop detectors that use automatic video image processing.

Stationary detectors usually detect single vehicles, and are capable to deliver:

- vehicle presence (time points when it enters and/or leaves the detection zone),
- vehicle speed,
- vehicle class (passenger vehicle, truck, bus, etc.),
- true or sensor-specific (e.g., magnetic) vehicle length.

Usually, the data gained from stationary sensors is transferred to a traffic management centre after being aggregated into intervals of 1, 5, 15, or 60 minutes. Non-aggregated real-time data is usually only used by traffic light controllers, including the occupancy of a detector and point speed measurements. The traffic light controller also derives other measures from this data like car following distance, occupancy percentage and queue counts. The non-aggregated data can also be sent to a central server for further processing to acquire more sophisticated measures. In Figure 2.1 the information sent to the server about traffic light state and detector occupancy is shown in a viewer. A thick blue bar represents an occupied detector and a line means unoccupied. For the signal heads, a thick green or yellow bar represent the signal head being green or yellow/amber, while a red line represents a red traffic light. When the data from a signal head and its associated detectors are combined, measures like saturation flow, waiting time and red light violations can be determined.

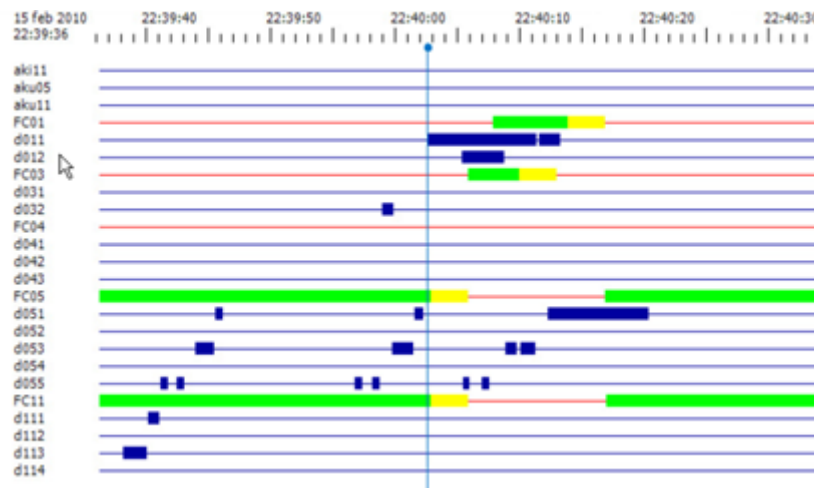


Figure 2.1: Traffic light controller log file.

The major feature of stationary sensors is the high quality of the obtained data and the high reliability. The major limitations of stationary sensors are the logistic and financial aspects. First, conducting a traffic monitoring study is expensive as it involves the installation of a sensor on or along the road. Second, if the intersection to monitor needs to be changed and adapted to varying traffic, it requires manual redesign. Finally, it might be difficult to extend the study of traffic correlation between multiple intersections or generalize it on a larger city area.

Another type of conventional traffic surveillance system is based on vehicle re-identifications. It is used to recognize a certain vehicle at several locations in the network. This technology is very intrusive, as it raises concern about privacy issues, but is efficient to measure travel time and to extract origin-destination matrices. The vehicle re-identification may be per se performed by various detection technologies such as:

- Automatic Number Plate Recognition (ANPR) – cameras capable of reading and recognizing license plates;
- Vehicle Inductive Profile (VIP) [Blokpoel, 2009];
- Short range wireless sensors (Bluetooth, NFC, WiFi);
- WiFi sensors.

2.1.2 Cooperative Traffic Surveillance

Besides ‘being detected’, conventional detectors do not require involvement of vehicles or drivers in traffic surveillance. Following the crowdsourcing trend, innovative traffic surveillance solutions could be developed based on the active participation of vehicles and drivers. The common concept, called ‘Floating Car Data’(FCD) is based on active position information obtained from vehicles, further processed by advanced data fusion and map matching techniques to extract traffic flows.

The first attempts at the beginning of this millennium proposed to use the active localization that mobile phone operators use to know in which cell their customers are. Although not relying on any GPS but using triangulations from multiple base stations, it is possible to extract some knowledge of the presence of a cell-phone on a specific road. The major limitation of this approach is the coarse precision of such information (30-100 m).

The large penetration of GPS-capable smart-phone and flat-rate data contracts with mobile phone operators gave birth to a far more reliable approach based on GPS location periodically transmitted to the cloud. In this concept, FCD operators require either a navigator or a GPS-equipped smartphone to periodically transmit their location, which it then anonymized and processed further. The main difference to the vehicle re-identification approach is the objective. Here, it is not the knowledge of one vehicle’s specific trajectory, but instead of extracting aggregated data about the

(past, present, future) amounts of vehicles on a specific road segment. The vehicle's trajectory – its progress through the road network over time – is computed by mapping individual position updates to a digital representation of the road network and connecting them. The information usually obtained from this procedure is the travel time at passed roads. Some other data processing chains exist that compute (or estimate) other information, for example the lengths of queues in front of traffic lights [Neumann, Wagner, 2012].

This approach is therefore less intrusive than keeping full trajectories, as most of the FCD operators only keep uncorrelated anonymous points, but it still requires FCD participants to be willing to provide this data (and bear the costs) as well as trust the FCD provider that their data will not be used for any other purposes.

As the data is sent via a mobile phone channel, the costs for such a system are relatively high, including both, purchasing a submitting device as well as a proper contract with a mobile phone operator. Due to this, such systems often re-use existing systems where possible¹ and are thereby often limited to dedicated vehicle fleets. This reduction to vehicles of a certain operation type (like taxis) but poses the problem of inherent deviations from usual traffic. Taxis, e.g., may on the one hand use bus lanes increasing their average velocity when compared to usual passenger traffic but may on the other hand move slower when looking for a new passenger.

Nonetheless, floating car data have proved to be a very valuable traffic surveillance system that in contrary to stationary sensors covers almost the complete network, especially when data collected in the past is additionally used.

Conceptually speaking, all FCD approaches are based on a similar three-step process

1. Information data sampling – retrieves large GPS data from users on the road;
2. Pre-processing and map-matching – filter data and map them on streets;
3. Model-based information estimate – from raw data, performance indicators (mainly travel time or average velocities) are derived.

TomTom's IQ traffic system deployed at 7 million TomTom users over 10 billion km of roads, is based on estimating the real mean speed on the road. One major innovation in TomTom therefore corresponds to using a flow 'model' that is based on historical real speed on the road rather than the maximum it should be. TomTom declares its capability to update its traffic data every 3 minutes.

Google is also providing FCD solutions with its Google Traffic application. Although little details are provided on its technology, it is to be expected that it follows similar concepts as TomTom. The major difference between TomTom and Google is the penetration rate. TomTom provides navigators, or applications that are open only when navigation is required. Google on the other hand, requires users to actively select to participate, regardless of using their maps or not. Yet, beside communication costs (to be supported by the phone owner), Google also offers the resulting image of the state of the road network for free, whereas TomTom generally requires a fee. Therefore, Google may reach a larger customer field than TomTom. A large variety of other FCD solutions exists, ranging from Cellint, AirSage, IntelliOne, ITIS holding, or media-mobile.

Beside the major companies, some FCD systems were developed by universities. The UC Berkeley built the "Mobile Millennium" application (<http://traffic.berkeley.edu/>). In a public-private partnership with Nokia Research Center and NAVTEQ, they provide a smartphone application that gathers GPS data, processes them and then return the result to the user. Similar to Google Traffic, it aims at showing an immediate and free benefit to users of their traffic information. Also similar to claims by TomTom and Google, the Mobile Millennium project anonymizes traffic data, in particular uncorrelates them from origin/destination points.

¹ The DLR operates some of such systems, based on disposition messages from taxis.

The French SINERGIT project aimed at improving the traffic information by complementing sampled GPS data by alternate sensing data, such as camera and fixed detectors. Similarly, the EU funded project Easy-OBUE demonstrated retrospective robust positioning of vehicles for those cases where no or insufficient signal reception from GPS navigation satellites was available. This was achieved by the combination of inexpensive inertial sensors with an innovative ‘non-causal filtering’ approach. For this purpose data was collected by the vehicle and transferred to a back-office, where processing could be carried out.

The major limitations of the current FCD may be summarized in three aspects:

1. **Cost** – Even if individually speaking, each vehicle only periodically transmits a few bytes of meta-data. Globally speaking, this requires a serious dimensioning of the communication network to support a city-wide monitoring. It is also expected that the cost of such dimensioning to be bearded by the mobile phone operators and transitively, by their customers.
2. **Privacy** – despite the privacy protection claims sworn by all FCD providers, the recent news still show that a lot may be learned from even anonymised data. Once such data reaches the cloud, there is no way a user can know how their data will be used or suppressed.
3. **Penetration** – TomTom and Google Maps do not share their community. So it is highly likely that their samples are also not shared, and TomTom or Google provide different views of the traffic conditions, which may only be a subpart of the full traffic.

2.1.3 Clustering and Topology Management

Considering all vehicles periodically transmitting data back to the surveillance server over cellular networks, a major issue is to use the available cellular capacity efficiently. One approach is to let vehicles transmit their data to a ‘leader’, which will be in charge of forwarding FCD from its followers over the cellular network. This has the advantage to reduce the number of connections and to optimize the transmitted resources per connection. However, this requires coordination between leaders and followers. Several studies proposed to let vehicles organize in clusters or other connected dominating sets (CDS) to off-load their cellular traffic on the cluster/CDS leader [Bazzi et al., 2012], [Stanica et al., 2013]. For instance, Ancona et al. [Ancona et al., 2014] investigated the cost on a communication system of large-scale FCD, and then evaluated the performance benefit from massively offloading FCD. They showed a 35% reduction in the uploading cost at no performance degradation. In general, the challenge behind this is to develop a cost-efficient topology to elect a leader that can sustain vehicular mobility.

2.2 Communication & Networking

2.2.1 Frequency Allocations

V2X Frequency Band in the US

In the US, the Federal Communications Commission (FCC) is responsible for spectrum regulation. In 1999, the FCC allocated 75 MHz of spectrum in the range 5.850-5.925 GHz (commonly called the 5.9 GHz band) for ITS, and specifically for the Dedicated Short Range Communication (DSRC) Service. In 2003, based on input from the ITS community and US Department of Transportation (DOT), the FCC issued licensing and service rules for the spectrum. These rules include a division of the spectrum into seven non-overlapping 10 MHz channels, and a 5 MHz unused band at the low end.

Each channel is further classified as either a Control Channel (CCH) or a Service Channel (SCH). Channel 178, in the middle of the seven 10 MHz channels, is the CCH. The other six 10 MHz channels are classified as SCHs, as are the two 20 MHz overlapping channels. The roles of the CCH and SCHs are not specified in detail yet, but one channel is reserved for safety-related

communications based on a Basic Safety Messages (BSM) containing the location and speed of the vehicle. Accessing these channels is so far restricted to WiFi devices supporting communications outside the concept of a Basic Service Set (BSS), abbreviated as OCB.

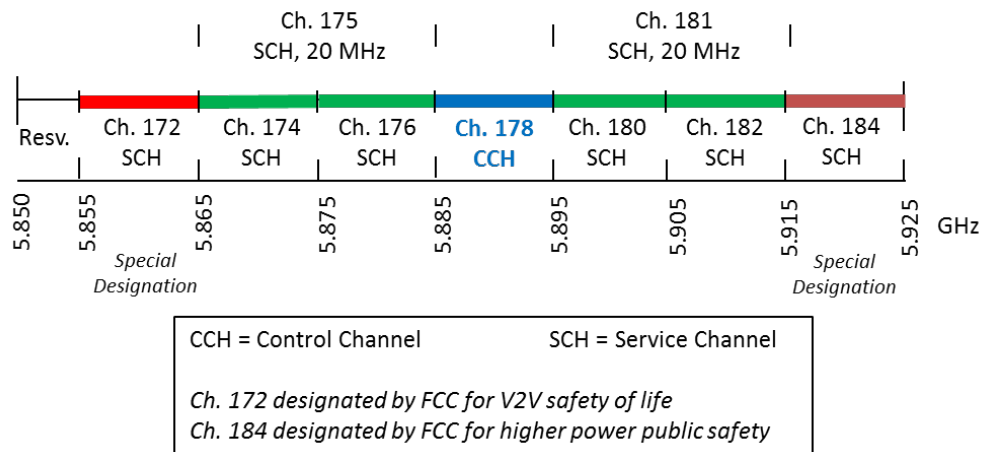


Figure 2.2: DSRC frequency allocation in the US.

V2X Frequency Band in the EU

In Europe, the ECC is responsible for spectrum regulation, and the European Commission (EC) is responsible to enforce that the allocated spectra are made available in all states of the EU. In 2008, an ECC Decision made 30 MHz of spectrum in the range 5.875-5.905 GHz (commonly called ITS-G5A) available for ITS restricted to safety-related communications, as well as an extra 20 MHz of spectrum in the range 5.905-5.925 GHz (commonly called ITS-G5D) for future ITS extensions.

Also in 2008, an ECC Recommendation made 20 MHz of spectrum in the range 5.855-5.875 GHz (commonly called ITS-G5B) available for ITS non-safety communications. Figure 2.3 illustrates the ECC frequency allocation plan and their different classes. As for the FCC spectrum allocation, the ECC allocation comprises of six SCHs and one CCH. Yet their EU-wide availabilities as well as their usage slightly differ from the FCC allocation.

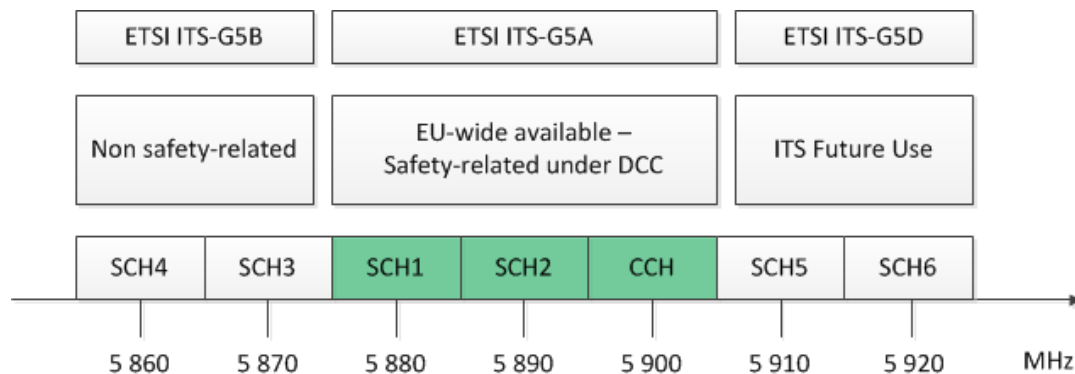


Figure 2.3: ETSI ITS-G5 frequency allocation in the EU.

As specified in [ES202663, 2009], the ITS-G5A frequency band contains channels CCH, SCH1, and SCH2, which are restricted to ITS road safety-related communications. SCH3 and SCH4 are contained in the frequency band ITS-G5B and are intended for ITS non-safety communications. Finally, SCH5 and SCH6 are part of the frequency band ITS-G5D and reserved for future ITS extensions. At the time of writing, the ITS-G5A band is the only ITS spectrum currently usable European-wide.

Via 2008 EC Decision [2008/671/EC], other ITS bands (ITS-G5B, ITS-G5D) have been allocated but not enforced to be made available. Depending on the states in the EU, these bands may or may not be available and usable.

One important difference between the US and the EU is that BSMs in the US are not sent on the CCH but rather sent on channel 172, a SCH specially designated for this purpose by the FCC, while the analogous CAMs and DENMs in the EU are sent on the CCH.

Another difference between the FCC and the ECC, is that although the ITS-G5A spectrum is restricted to safety-related communications, it is not restricted to a specific technology [ECC/DEC/(08)01, 2008/671/EC]. In principle, if other technologies than ITS-G5 (e.g. 3GPP LTE, WiFi-Giga) could operate in 10MHz OFDM channel for traffic safety, they could operate in the ITS-G5A band.

2.2.2 Investigated Technologies

Dedicated Short Range Communication (DSRC) / ETSI ITS-G5

Dedicated Short Range Communication (DSRC) appeared in the US 1999 as a communication technology adapted to dedicated communications between vehicles and road infrastructures (in particular toll booth) at 900 MHz. Initially based on a RFID technology, a IEEE 802.11 WiFi technology replaced it later to increase capacity and range. The DSRC technology has later been moved to the 5.9 GHz in 2003 for safety-related Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications.

In Europe, DSRC is named ITS-G5 (by the ETSI) in order to avoid confusion with the other toll system technology CEN DSRC equipping trucks. Similarly to DSRC, ITS-G5 is based on a WiFi technology adapted to vehicular environment and operating since 2008 at 5.85 GHz to 5.905 GHz frequency band. DSRC also exist in other countries, such as in Japan under the name URIB DSRC and is operating in 5.77-5.85 GHz since 2001.

The common aspect of all DSRC technologies for V2X communications is that they are based on a vehicular extension to the IEEE 802.11-2012 standard. Initially known as IEEE 802.11p, the vehicular amendment to IEEE 802.11-2007 extends the IEEE 802.11a WiFi by adapting it to the highly dynamic vehicular environment at 5.9 GHz. The differences between IEEE 802.11a and IEEE 802.11p may be described from a PHY and MAC layer perspective.

At the physical layer (PHY), high vehicular mobility makes vehicular communication subject to Doppler effects, which may then create OFDM Inter-Symbol Interference (ISI). To mitigate this, the OFDM guard interval has been doubled, which in turn also doubles the OFDM symbol time and by consequence, reduces the channel bandwidth by half. Accordingly, both the US FCC and the EU ECC assigned IEEE 802.11p to a 10 MHz bandwidth rather than the 20 MHz WiFi bandwidth. As consequence, the minimum and maximum coding rate of the IEEE 802.11p ranges between 3 Mbps – 27 Mbps instead of 6 Mbps – 54 Mbps.

At the media access control layer (MAC), vehicles cannot first authenticate and negotiate with other vehicles or stationary road side units to join a Basic Service Set (BSS). The high dynamic vehicular environment requires vehicles to be able to spontaneously and directly communicate “outside the context of a BSS” (OCB). This aspect gave the name of the IEEE 802.11p [IEEE 802.11p-2010]. Accordingly, from a MAC perspective, vehicles operating a WiFi supporting the OCB mode are allowed to transmit and accept packets from other vehicles outside of a BSS. The authentication is implicitly assumed by using signature and asymmetric cryptography, and controlled by upper layers functions.

Since 2012, the IEEE 802.11p has been integrated into the latest IEEE 802.11-2012 standard [IEEE 802.11-2012], and the IEEE 802.11p should now be referred to as *IEEE 802.11-2012 OCB at 5.9 GHz over 10 MHz*. The complexity of this denomination makes that community rather refer to either the US or ETSI standard denomination, *DSRC* or *ETSI ITS G5* respectively. Table 2.1 provides the key differences between WiFi and DSRC.

Table 2.1: Functional Differences between Wifi and DSRC.

	WiFi-5	DSRC / ETSI ITS-G5
Frequency	5.4 GHz	5.9 GHz
Bandwidth	20 MHz	10 MHz
Max Transmit Power (max)	23/30 dBm EIRP	33 dBm EIRP
Range (max / expected)	200 m / 50m	1000 m / 300 m
OFDM symbol time	4 μ sec	8 μ sec
Data Range	6-54 Mbps	3-27 Mbps

IEEE WAVE / ETSI ITS

The DSRC technology per say is not limited to the access layer, but also includes higher layers protocol stack to support V2X communications. In the US, it is known as Wireless Access in Vehicular Environments (WAVE), also known as IEEE 1609. In Europe, the ETSI simply named it the ITS station stack.

The IEEE 1609 family of standard is depicted on Figure 2.4. The lower layer (PHY/MAC) correspond to the DSRC access technology (multiple transceivers are possible). The higher layer corresponds to V2X applications, such as traffic safety or to the COLOMBO traffic surveillance applications. The particularities of WAVE are shown at the middle layer. Although still containing an IPv6 protocol stack, WAVE also include a non-IP stack called Wave Short Message Protocol (WSMP), described in IEEE 1609.3. Although IPv6 remains critical for global reachability, IP is neither required nor mostly efficient for short range dedicated connectivity, especially for one-hop communications. Accordingly, WSMP specifies network headers using a MAC 64-bit identifier type rather than an IP address. The specification of the addressing and the related security aspects are described in IEEE 1609.2. Also, vehicles supporting WAVE may also communicate on multiple channels. Multi-channel switching mechanisms are described by WAVE in 1609.4.

Beside the protocol stack, the most important aspect of WAVE are the supported applications. Currently, two major applications are envisioned. Traffic Fee Collection, described in IEEE 1609.11, use dedicated messages exchanged between vehicles and stationary communication units (called Road Side Unit (RSU)) to deal with financial transactions related to traffic fees.

Traffic Safety Application, most probably the most critical and ambitious WAVE application so far relies on a specific message called Basic Safety Message (BSM) to make vehicles 'aware' of the presence of traffic danger. BSM are not only unique from their usage, but also both from their content and their transmit strategies. BSM contains a header (specified by SAE J2735) containing configurable GPS location data, as well as traffic context codes. Second, as the primary objective is to let other vehicles be aware of one's vehicle driving dynamics (position, speed), BSM are periodically transmitted at a rate varying between 1 Hz to 10 Hz.

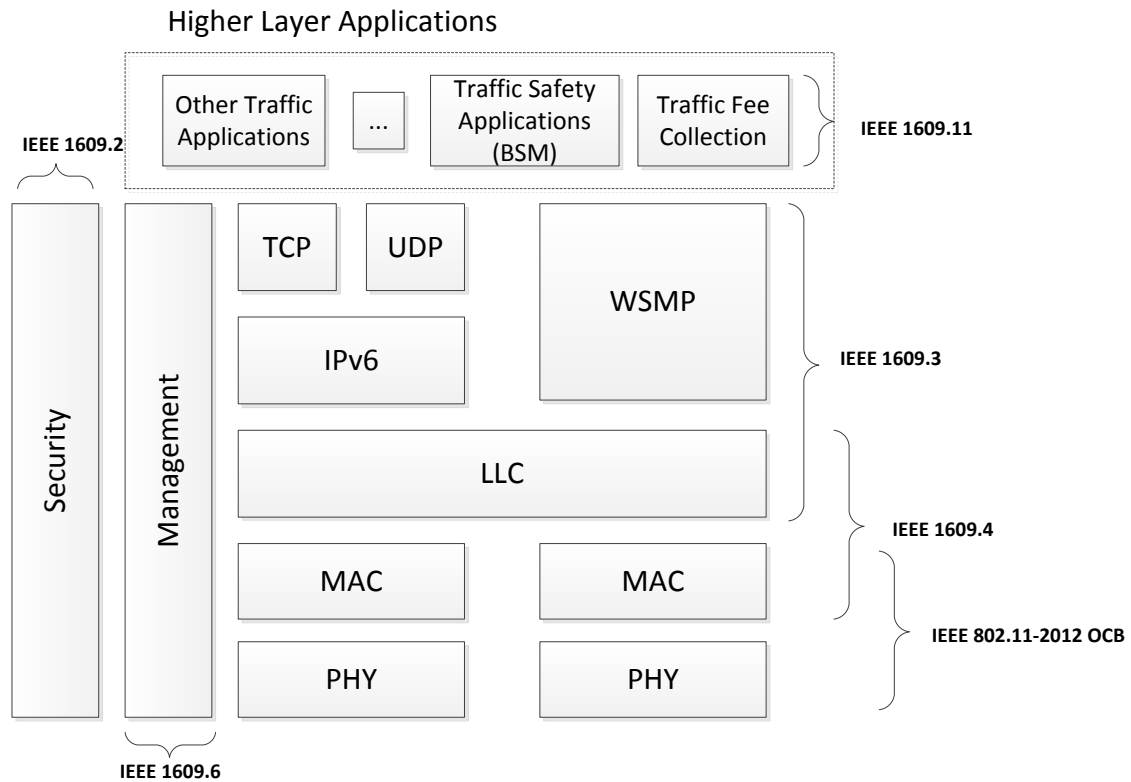


Figure 2.4: IEEE 1609 Family of Standard and the WAVE protocol stack.

Similarly to WAVE, the ETSI also provides a full vehicular communication protocol stack on top of ITS-G5. As illustrated in Figure 2.5, the ETSI ITS stack bears similarities with WAVE, but differs in two particular aspects. First, whereas WAVE only provides headers for WSMP, and no specific networking details, ETSI provides network-level protocols based on geographic information. Called Geonetworking, the ETSI networking stack is also non-IP and provides network-level headers, as well as various geographic routing protocols, such as greedy routing or contention-based forwarding.

The second important different with WAVE is the Facilities Layer proposed by the ETSI. It is considered as a service layer to higher layer applications, and includes some critical services for traffic safety applications. First, it is responsible for the transmission and processing of the *Cooperative Awareness Message (CAM)* [CAM-ETSI, 2013], the ETSI equivalent of the WAVE BSM. The content of CAM headers are similar to BSM, as they are both specified by SAE J2735, and are also periodically transmitted at a rate between 1 Hz and 10 Hz. CAM are never forwarded on multiple hops. When a traffic-related information requires to be reached by vehicles farther than the communication range, *Decentralized Environmental Notification Messages (DENMs)* [ETSI-DENM, 2013] are used. DENM can be routed via geographic unicast mechanisms over multiple hops to reach a given area, and then via multi-hop broadcast mechanism to reach all vehicles in that geographic area.

A third type of message, equivalent to the CAM but only send by stationary RSUs, are called *Signal Phase and Time (SPaT)* messages². They are mainly and primarily used for traffic lights to transmit TLC-related data to vehicles. SPaT are so far assumed not to be relayed, but it could be envisioned that would be, in the case coordination between multiple traffic lights would become required. SPaT

² See also [COLOMBO D4.3, 2014], section 3.2.2

messages also reach farther vehicles compared to CAM mostly due to RSUs higher antenna heights, and consequently communication range.

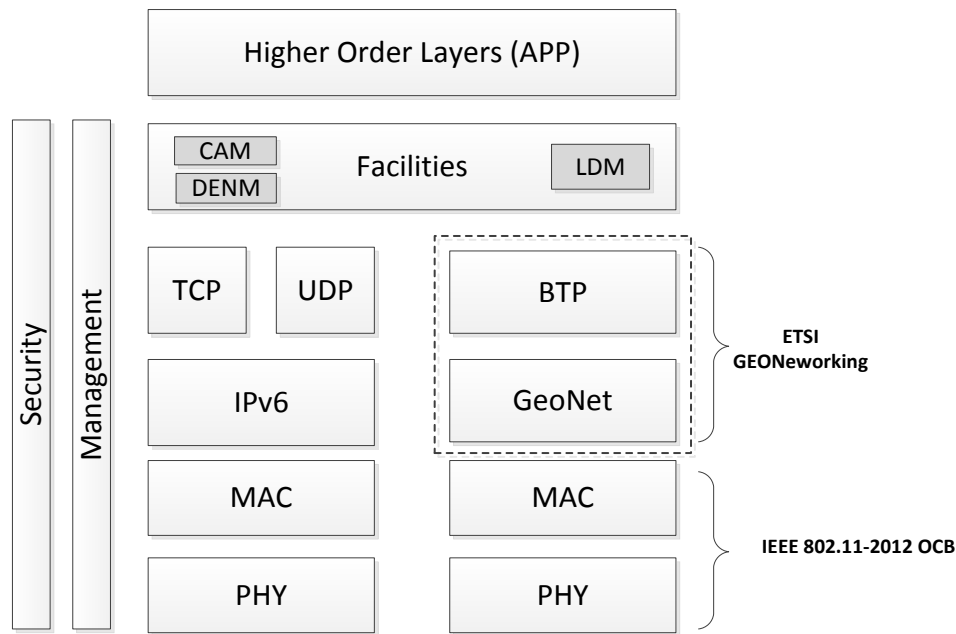


Figure 2.5: ETSI ITS Station Architecture.

From a COLOMBO perspective, CAM/BSM messages are a very good option for COLOMBO's traffic surveillance solutions, as the safety-related awareness (position and speed of all neighbouring vehicles) can be directly used as input to local traffic monitoring. Also, as BSMs/CAMs are transmitted by default, it is traffic related information that can be used at no extra overhead.

When traffic surveillance information need to be transmitted to the traffic light (over a RSU), vehicles may either directly reach the RSU over unicast if it is in their direct range, or benefit from the ETSI standard geographic unicast standard to disseminate traffic surveillance data to a RSU outside of their communication range.

COLOMBO may also rely on SPaT messages to notify vehicles of traffic surveillance duties. It should yet be mentioned that the link between a RSU and a vehicle will mostly be asymmetric, as a vehicle may receive a SPaT, but the RSU might not receive CAM from that vehicle.

Table 2.2 summarize the key information, freshness and range that can be obtained from one hop and multi-hop V2X communication.

Table 2.2: V2X Message Properties.

Channel	Message	Information	Distance	Range
V2X-OBUE	CAM/BSM	GPS position/speed	1-hop	300m
V2X-OBUE CAM	DENM	GPS position/speed & configurable container	m-hop	>> 300m
V2X-RSU	SPaT	TLS info & configurable container	1-hop	1000m

2.3 V2X-based Traffic Surveillance Approaches

2.3.1 V2X communication-based

Beside the traditional traffic safety applications, another major traffic-related class of applications that could greatly benefit from V2X communications is Traffic Information Systems (TIS). It is expected that vehicles being inside traffic are the perfect sensors for traffic-related information. V2X brings the advantage of allowing local processing and distributed traffic gathering.

V2X TIS may be decomposed in three challenges: 1) obtaining local traffic measurements, 2) consolidating data, and 3) transporting data. Although a large variety of related work addressed V2X data dissemination independently to an application, the specific type of data gathered by TIS often requires a tight coupling between local data consolidation and data dissemination strategies.

On the first challenge, vehicles are mobile telematics platforms, which naturally need to gather proximity traffic states. Vehicles supporting V2X communications have a more robust navigation system. Considering the constant need of a precise lane-level positioning, vehicular navigation systems are not only based on GPS signal, but also enhanced with complementary data from various sensors in vehicles, as well as robust information fusion engines capable of matching a vehicle on a road and on a specific lane, even in the case of temporary GPS-signal loss. Compared to smartphone or 'bare' GPS receivers, vehicular navigation systems therefore provide a higher position and speed quality, as well as detailed map data providing abstract position information, such as intended driving direction, road ID and lane type (regular lane or turning lane).

According to both standards, WAVE and ETSI ITS, this precise position information is periodically shared between vehicles via V2X communications. Accordingly, each vehicle has a Location Table keeping records of the location and traffic dynamics of its neighbors. Considering potential congestion from transmitting all these information, CAMs and BSMs only require GPS position and speed to be compulsorily transmitted. Yet, CAMs/BSMs are flexible and can easily be accommodated to transmit some extra information from the vehicular navigation system as a function of the ITS requirements.

There are several approaches that propose to benefit from CAM/BSM data for traffic surveillance. For example, Jerbi et al. [Jerbi et al., 2007] proposes to let vehicles elect a group leader according to their mutual distance, and then let the group leader gather and compute local traffic information. The group leader later transmits this data to other group leaders and to the traffic lights. The CoTEC distributed cooperative traffic surveillance system [Gozalvez, 2010] also proposes to rely on periodic CAMs to guess the density and speed of the neighboring vehicles. They use fuzzy logics based on estimated traffic in different Level-of-Services (LoS) to map the density/speed to the occurrence of a traffic jam.

When CAM/BSM either cannot be used or require to be modified for the purpose of the TIS application, Akhtar et al. [Akhtar et al., 2012] adapted P2P discovery techniques and proposed three traffic density estimation techniques. The first one called *Sample&Collide* adjust the transmit time to maximize vehicles being detected at the lowest overhead. The second called *HopSampling* manage to estimate the traffic density from the hop counts of specific probe messages. Finally, *Gossip-based aggregation*, where adapt the selection of the random neighbor to send a P2P probe. It seems that simple hop probing already manage to provide a good guess. However this study has been only mainly performed on stable flow on highway conditions.

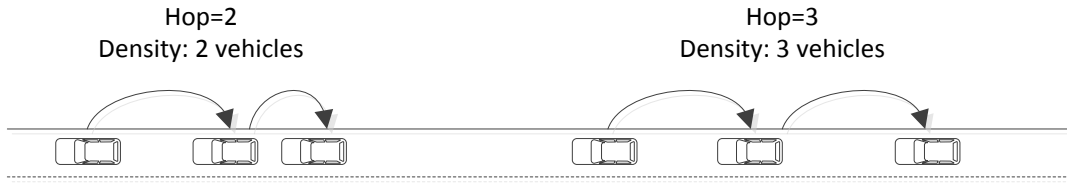


Figure 2.6: HopSampling Traffic Surveillance - density is related to # hops of active probes.

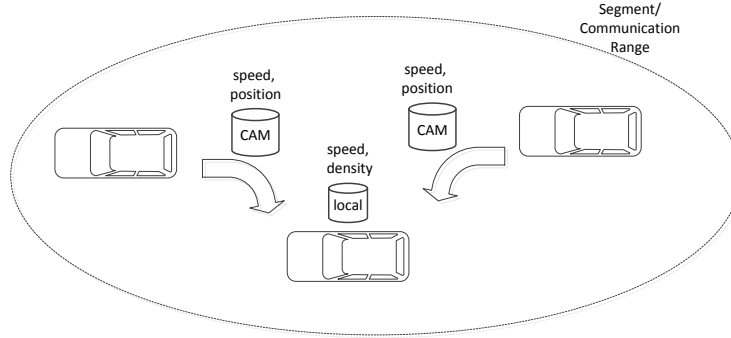


Figure 2.7: CAM-based Traffic Surveillance – Vehicle receives GPS position & speed from neighbours – density is related to the number of neighbour in range/segment.

Considering the large amount of vehicles and city range, it is also highly unlikely to let all vehicles to simply periodically transmit the content of their Location Table over the vehicular network to all recipients who might be interested due to bandwidth limits. Accordingly, vehicles need to locally consolidate data through V2X direct communications, also known as data ‘summarization’ or ‘aggregation’. The basic mechanism would be to average multiple sources of same data. Approaches proposed by Xu and Barth [Xu, Barth, 2006] use a time-stamp based moving average. Every data is sent with a sampling time stamp to all vehicles, which then aggregate them using a moving average:

$$m(t + T) = \frac{1}{n+1} (n \cdot m(t) + s(T)), \quad (2-1)$$

where $s(T)$ represents a new sample obtained at time $T > t$, $m(t)$ the moving average value at time t , and n is the current number of samples. Considering a measure of speed, such approach would therefore require every vehicle to periodically transmit three fields: $s(T)$, n and T for neighboring vehicles to be able to integrate them.



Figure 2.8: Segment-based Traffic Surveillance (e.g. SODAD).

Another variant of a time-stamped averaging scheme is called segment-based averaging and is proposed in SOTIS [Wischhof et al., 2003b], [Wischhof et al. 2005]. The general scheme is called ‘Segment-Oriented Data Abstraction and Dissemination (SODAD)’, and is based on a sub-division of roads into shorter road segments. Individual information is shared between vehicles, which then consolidate them with data from the same road segment. SOTIS and SODAD provide speed

averages either on the full road (SOTIS) or on the road segments (SODAD). The major drawback of this approach is that the road segments are hard-coded and cannot be adapted to dynamically changing conditions.

Instead of combining traffic information within a road segment, other approaches propose to combine information within a cluster of vehicles. The Zone Diffusion Protocol [Bronsted, Kristensen, 2006] divides roads into ‘cells’ and proposes to merge observation data on the same cell according to an application-specific data combination. An example of such application-specific combination is for instance icy road, where if at least one cell indicates an icy road, then the whole road is considered icy.

Another approach followed by Traffic View [Nadeem et al., 2004] proposes to combine data within clusters to transmit position and speed information corresponding to a group of vehicles, rather than each vehicle. Accordingly, TrafficView proposes an aggregation ratio, which provides the number of required vehicles to consolidate data. This ratio clearly depends on the speed/position correlations between vehicles in the cluster, as well as the local traffic conditions and road type to keep aggregation errors as small as possible.

The CASCADE scheme [Ibrahim, Weigle, 2008] is similar to TrafficView, but uses syntactic information compression within a cluster. Instead of computing a mean value of traffic information from all vehicles within a cluster, CASCADE computes a centroid position of the cluster, stores and then shares only relative data from the mean cluster information. The major benefit of this approach is that relative values of individual vehicles in a cluster vary less than the absolute values, and can also be transmitted with less overhead.

Although not based on a direct V2X-based traffic data exchange (or consolidation), StreetSmart [Dombush, Joshi, 2007] proposes an interesting data consolidation technique, which is specially tailored to not report mean values of traffic information but instead deviation (and unexpected) values from the mean. This is typically important to quickly detect traffic jams, or other traffic anomalies. Streetmap lets vehicles record their movements and relies on data mining to generate an abstract view that shows how close or far this value is from an expected movement. These values are then sent to the network, which then aggregates it to extract a group of samples with similar patterns.

2.3.2 FP7 iTETRIS

The FP7 ICT iTETRIS project was a project co-financed by the European Commission under the 7th Framework Program between 2007 and 2010. It grouped a consortium composed of experts in V2X communications, simulations, vehicular mobility and simulation from 5 countries (DE, FR, IT, SP, NL), three of them being also in the COLOMBO consortium (DLR, EURECOM, PEEK). The iTETRIS project aimed at developing a long-term, global, sustainable, and open vehicular communication simulation platform facilitating large scale, accurate, multidimensional evaluation of cooperative ITS solutions for traffic efficiency. We describe in the section the major highlights of the iTETRIS project that are related to COLOMBO objectives.

Simulation Platform and ITS Traffic Efficiency Applications

The iTETRIS project’s major objective was to develop a large scale open-source ITS simulation platform. The resulting system is composed of three modules, a network simulator (ns-3), a traffic simulator (SUMO), and an ITS application module, altogether federated by the iTETRIS Controlling System (iCS). The iTETRIS platform is used and extended by the COLOMBO project, and is described with greater details in [COLOMBO D5.1, 2014].

Another objective of iTETRIS has been to validate the benefit of the iTETRIS platform by developing six traffic efficiency ITS applications as depicted in Figure 2.9.

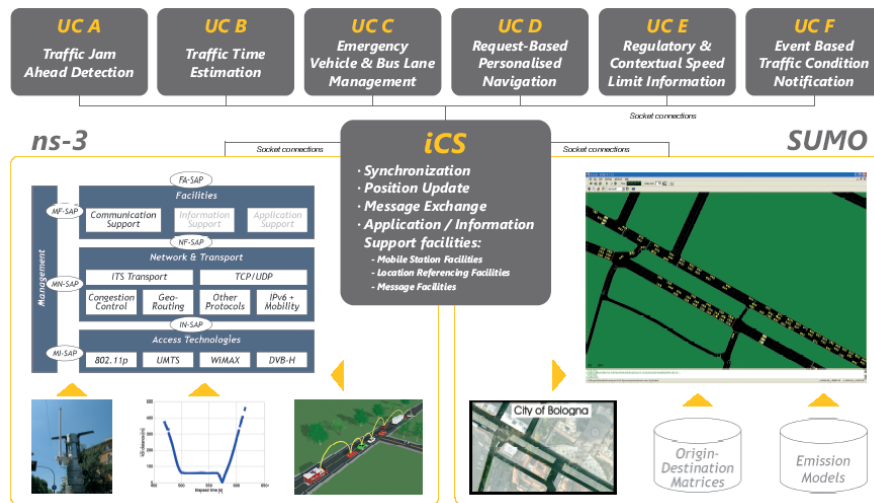


Figure 2.9: The iTETRIS Simulation Platform, including the six ITS applications From [iTETRIS].

Among them, two applications are related to distributed cooperative traffic surveillance, and as such are related to the objectives of COLOMBO:

- *Traffic Jam Ahead Detection* - It aims at relying on cooperative ITS communications for vehicles to detect traffic congestions.
- *Traffic Light Adaptation* - Based on traffic data from the traffic Jam Ahead Detection, or the Travel Time Estimation, it can adapt the traffic light policies of the selected intersections and evaluate its impact based on its potential reduction of traffic congestions.

The iTETRIS project also provided several traffic scenarios (urban, highways), which have also been used to conduct V2V and V2X connectivity evaluations, and optimize the placement of RSUs.

Connectivity and Impact of Infrastructure

The objective of the connectivity analysis conducted by iTETRIS was to evaluate the graph properties of a multi-hop V2X network. The impact of various penetration rates, as well as benefits of different RSUs locations and numbers have been evaluated.

On Figure 2.10, the number of connected components is illustrated. A connected component is defined as the largest set of vehicles which are connected (via one-hop or multi-hop). Such components are by design unstable due to vehicular mobility, but the evaluation of a stability value could be critical to the design of V2X-based traffic surveillance, as well as infrastructure planning.

The connectivity results conducted during iTETRIS only assumed vehicles and V2X technology. COLOMBO assumes the availabilities of alternate technologies, such as WiFi-Direct, which could help compensate the low penetration. Also, pedestrians can also bridge the gap between two vehicular connected components. These aspects are worth investigating within COLOMBO.

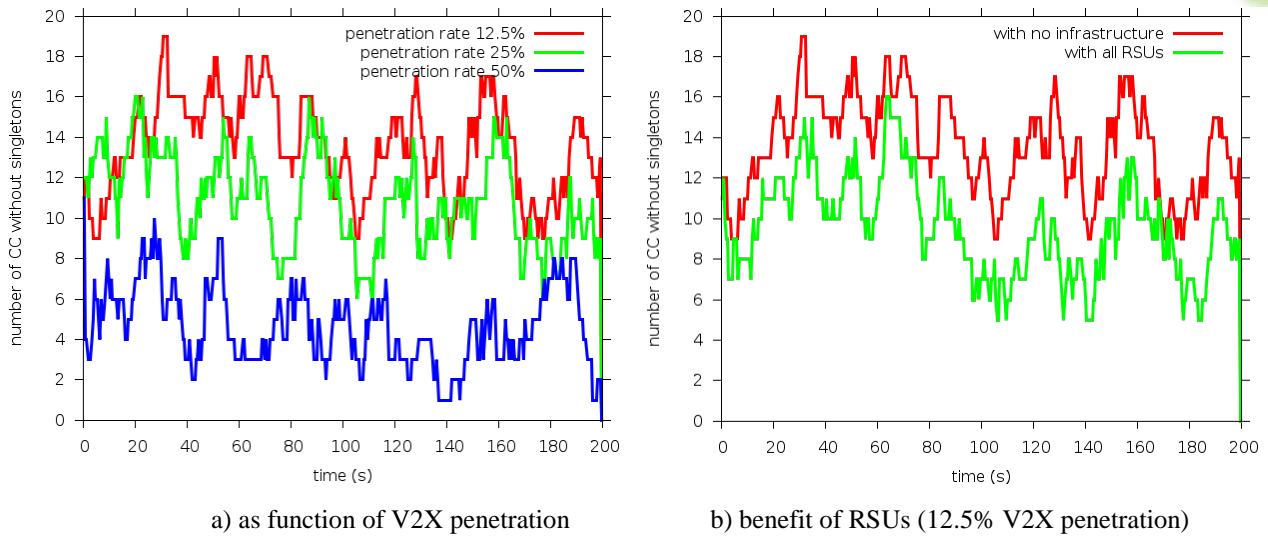


Figure 2.10: Number of connected components in an urban vehicular network scenario. From [iTETRIS].

V2X Traffic Surveillance

iTETRIS has proposed and evaluated a novel cooperative technique based on Vehicle-to-Vehicle (V2V) communications and fuzzy logic to detect road traffic congestion in highway scenarios without the need to deploy infrastructure sensors. The proposed technique is also capable to accurately detect the traffic congestion intensity and length. Basically, each vehicle estimates its local traffic conditions based on its vehicular speed and surrounding traffic density. The local traffic density is estimated through the reception of CAM messages from neighbouring vehicles. Based on the speed and density estimates, each vehicle by means of a fuzzy logic system is able to locally detect a potential road traffic congestion condition. Once a traffic jam is locally detected, a cooperative procedure based on multi-hop communications is activated to achieve a consensus decision on the traffic congestion situation. The proposed approach allows collaboratively evaluating the individual estimations that different participating vehicles make locally.

Figure 2.11 depicts the traffic jam detection capabilities of cooperative CAM-based V2V communications. The figure shows the creation of a traffic jam (from 300 s, peak at 600 s), its lengths (2000 m), as well as its intensity (blue to red).

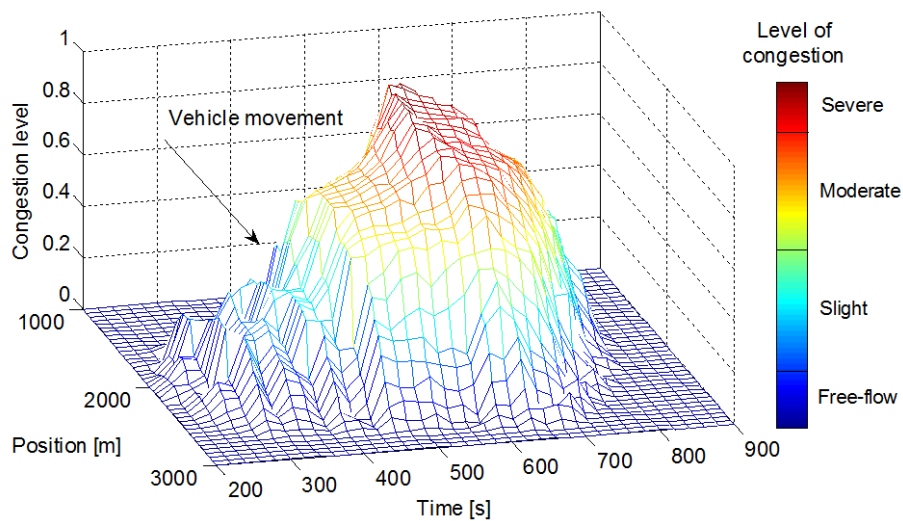


Figure 2.11: Traffic congestion local estimation based on V2V communications. From [iTETRIS].

Figure 2.12 shows the iTETRIS CoTEC decentralized mean speed and density on a highway section. It also put the density in perspective to the expected Highway Capacity Manual benchmark of moderate and severe congestion to provide an abstract ‘Congestion Level’ rather than an absolute value.

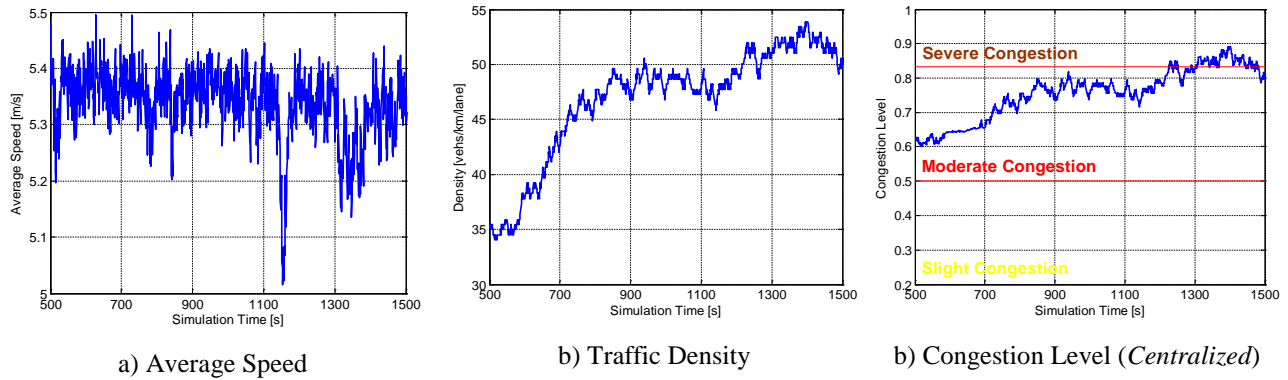


Figure 2.12: Traffic parameters and congestion level on a highway scenario. From [iTETRIS].

Figure 2.13 then illustrates the benchmarking of the iTETRIS Traffic Jam Ahead Detection (CoTEC) compared to a ground truth provided by the traffic simulation SUMO. As it can be seen, the CoTEC solution slightly overestimates the ground truth, although the authors later provided optimization algorithms to fit better to the expected conditions.

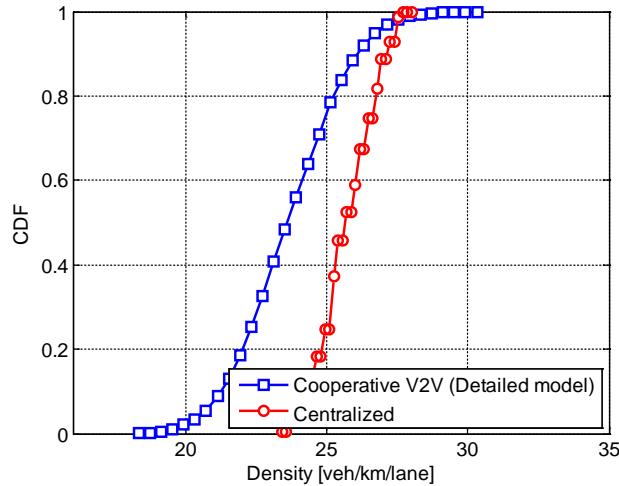


Figure 2.13: Cumulative Distribution Function (CDF) of the traffic density estimation; Cooperative V2V and Centralized detection techniques From [iTETRIS].

The traffic surveillance solutions showed to be capable of detecting traffic congestions very efficiently and at very little overhead (as it is based on CAM). However, this solution has been tested only on highways, as it still requires knowledge of stable flows (expected mean speed) in various congested conditions. In COLOMBO, our objective is to develop traffic surveillance adapted to dynamic flow conditions. The iTETRIS algorithm still remains an approach that may be investigated in the COLOMBO framework.

3 Scenarios and Use Cases

3.1 Traffic Scenarios

The V2X traffic surveillance algorithms developed by COLOMBO will be evaluated either on synthetic urban roads, or two intersections extracted from the urban mobility scenarios in Bologna.

3.1.1 Synthetic Intersection Scenario

During the development of the V2X traffic surveillance solutions, a simple and controllable scenario is required to adjust the behaviours. Accordingly, a set of two synthetic scenarios are used, which correspond to a simple Cross-shape intersection with either one lane or two lanes in each direction.

The traffic volumes are also artificially reproduced to correspond to two situations: stable traffic flows, and increasing traffic flows. The former is used to test if the V2X solutions are estimating correctly a stable traffic flow, and the latter is used to evaluate if the V2X solutions are capable of detecting an increasing flow.

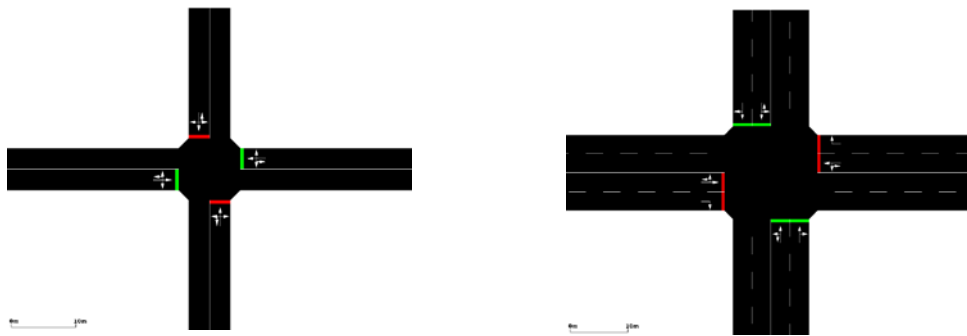


Figure 3.1: Synthetic Intersection Scenarios.

3.1.2 Bologna Urban Scenarios

For more realistic evaluation of the performance of the V2X traffic surveillance solution, two real intersections are used from the Bologna Pasubio-Costa Joined scenario. More details related to the global scenario are provided in D1.1.

The traffic conditions correspond to increasing traffic flows between 8:30 and 9:00. Figure 3.3 illustrates the two scenarios, including configuration parameters, such as the various directions of the incoming/outgoing flows, as well as the number of lanes.

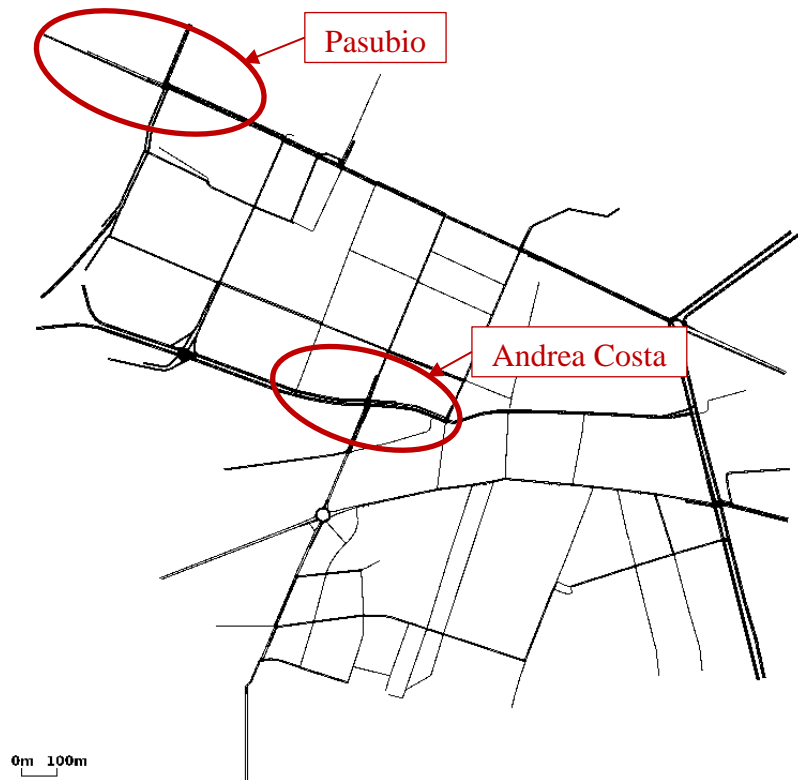


Figure 3.2: Bologna urban 'Joined' Scenario, and location of the two intersections.

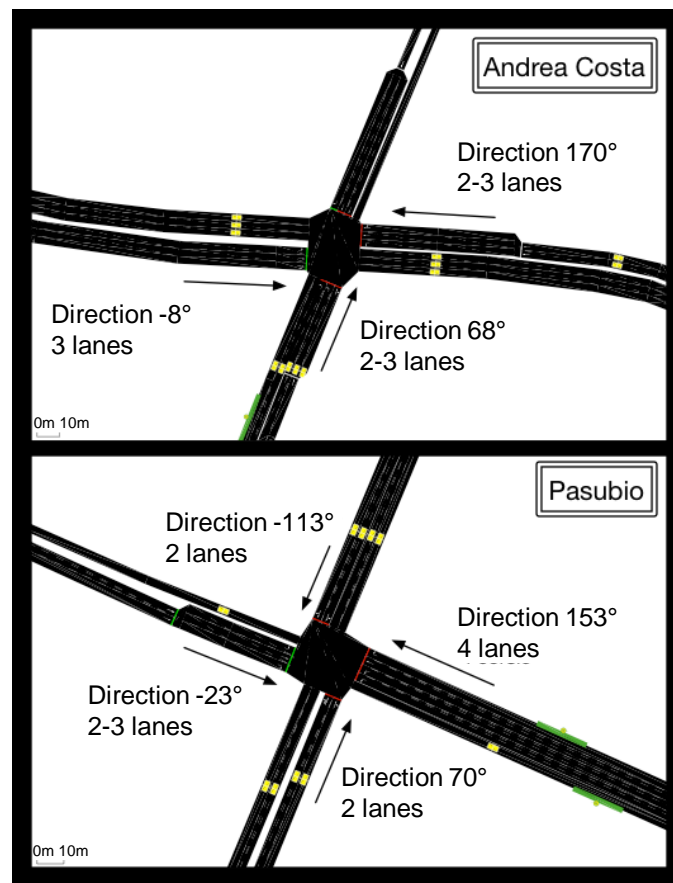


Figure 3.3: Two Urban Intersection Scenarios in Bologna used for the evaluation of the COLOMBO traffic surveillance algorithms.

3.2 Traffic Light Scenarios

3.2.1 Self-Organizing Traffic-Light Control System

Most of the related work in cooperative traffic surveillance does not provide explicitly the impact on traffic lights. They provide traffic density/speed estimations without specific application in mind. In fact, for the same reason that V2X communication patterns are connected to the need of traffic surveillance parameters and the underlying vehicular mobility, traffic surveillance parameters depends on the specific requirement of a specific traffic light control algorithm and the current vehicular flow model. This tight coupling makes traditional traffic surveillance be adapted for a specific intersection and for a specific expected flow. COLOMBO's proposed adaptive traffic light control system will adapt to dynamic traffic, and as such may also adapt its traffic surveillance requirements.

COLOMBO TLC has two major innovative aspects: first it aims at being able to dynamically adapt to changing traffic conditions. Second, it also aims at obtaining its traffic flows from a DFCD traffic surveillance system. Finally, it will also aim at easing the synchronization between traffic lights. For instance, COLOMBO TLC system will not allow a green light to a street segment it knows it is already congested. Accordingly, considering a typical X-shape intersection as depicted in Figure 3.4, COLOMBO TLC will not only require incoming traffic flows, but also outgoing traffic flows.

Although TLC are developed in WP2, a tight interaction has been established between WP1 and WP2 in order for the TLC to provide the required data they need, and the Traffic surveillance systems to provide that kind of data at the required quality usable by TLC.

From WP2, data required by COLOMBO TLC are summarized as follows. First, on the incoming road, TLC would require to have aggregated knowledge of speed/density for each lane. Aiming at providing a similar functionality as regular sensors, TLC may require traffic information at varying distances (i.e. multiple zones). The closest one being critical to detect high/low flows, farer zones may also be beneficial for the TLC to predict the evolution of the closest zone, and as such, anticipate its strategy.

Second, COLOMBO's TLC also requires the knowledge of speed/density for each exiting lane in each direction of the traffic light. Although multiple zones might also be beneficial, the first exiting zone is expected to be already sufficient.

The size (length) of the incoming and outgoing zones may be adapted to the traffic conditions. Optimally speaking, traffic flows within one zone should be stable in order for the traffic surveillance to provide a meaningful time-sampled moving average. So, depending on the flow dynamics, COLOMBO's TLC may increase the number of zones, or increase/decrease their lengths. As COLOMBO's traffic surveillance does not rely on static sensors, it can easily be adapted to the new requirements.

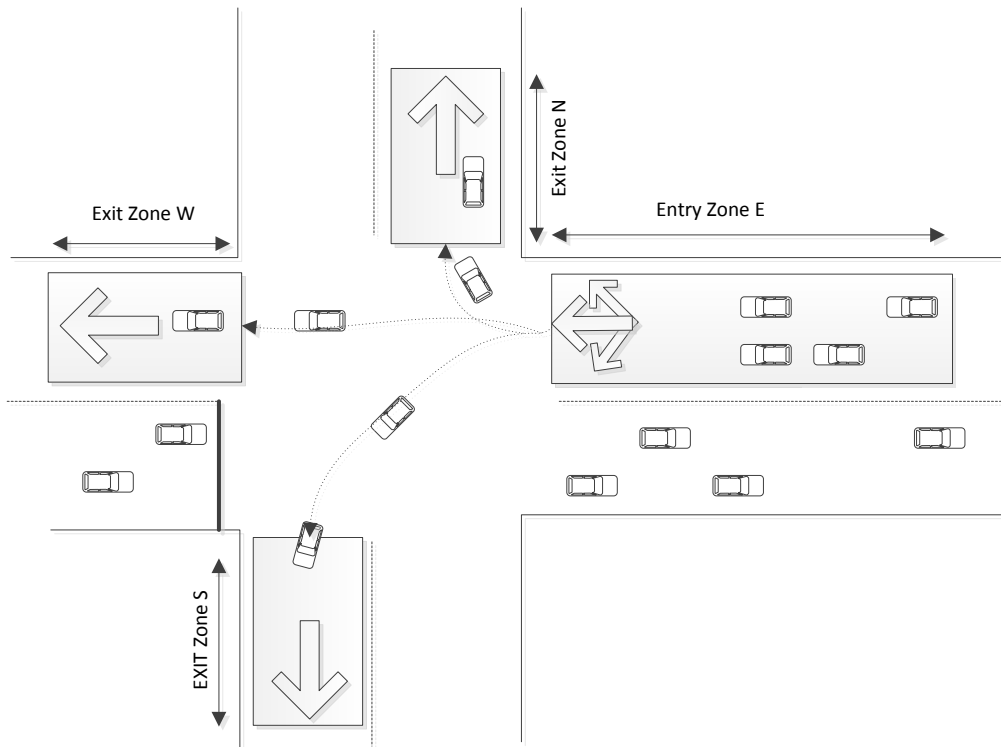


Figure 3.4: Traffic Surveillance Zone required by Traffic Light Control - One Entry zone, band three Exit zones.

The number and size of each zone, as well as the required data quality is expected to be provided to all communication devices participating to the V2X traffic surveillance system by a RSU.

3.2.2 Virtual Sensors

As described, COLOMBO's Traffic Light Control system will require traffic surveillance data within different geographical zones of variable lengths. COLOMBO's TLC is also built to be compatible to traffic data from traditional stationary sensors. Accordingly, V2X traffic surveillance data should provide data following a similar semantic and structure as regular sensors for a smooth integration in COLOMBO TLC. COLOMBO therefore proposes to rely on the concept of 'virtual sensors'.

Virtual sensors are deployed at a location where the TLC would require traffic samples. It is not only an overlay structure responsible to coordinate data sampling similar to the road segments as proposed in SODAD. It is also a structure that will organize data quality, timing and naming, which means that traffic data provided by a stationary sensor or a virtual sensor will have the same meaning.

But the major advantage of virtual sensors is that they can dynamically change their location, their size, the type or quality of the sensed data as a function of the dynamic change of traffic flows and the requirements of the traffic light control. Considering Figure 3.5, stationary sensors are capable of providing traffic data at immediate range from the intersection. But if the TLC requires traffic data at other locations, new stationary sensors would need to be deployed, which would be particularly inefficient if the traffic data is not always required at this distance. Or, the TLC deploys virtual sensors at the required locations and notifies all approaching vehicles via V2X communications. Accordingly, as illustrated on Figure 3.5, the TLC is capable of obtaining extra traffic data where stationary sensors may not be deployed.

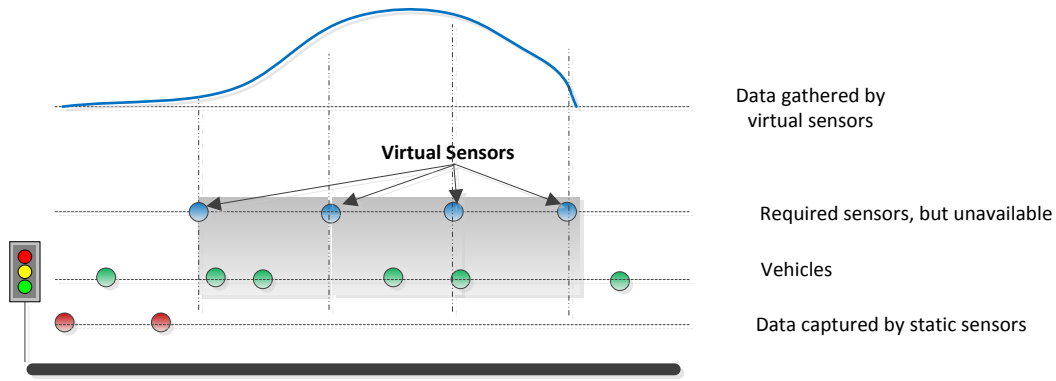


Figure 3.5: Virtual sensor concept, where extra virtual sensors are deployed by COLOMBO's TLC where it requires traffic data.

Virtual sensors in COLOMBO are therefore a container including data as provided on Table 3.1. The required data are periodically transmitted by the TLC to all vehicles approaching its intersection via a V2X specific message, such as the ETSI MAP message. MAPs are periodically transmitted (1 Hz – 0.2 Hz) by a TLC and indicate the precise geometry of the intersection. The location and requirements of the virtual sensors may be appended in a generic container. Accordingly, approaching V2X-equipped vehicles are always aware of the location and data required by all virtual sensors when approaching an intersection.

Table 3.1: Data required by virtual sensors.

Type	Values/Meaning
Road IDs	Ordered list of road IDs included in the virtual sensing zone
Lane IDs	Ordered list of lane IDs per each road ID included in the virtual sensing zone
Position	Segment $[D_{min}, D_{max}]$ corresponding on the Euclidian distance from the TLC from where the virtual sensor starts, resp. ends.
Data	Ordered list of data to be sampled by the virtual sensors (e.g. speed, density)
Rate	Sampling rate required by the TLC
Flow	Traffic flow to be monitored (turning, direct, contra-flow..)
Restrictions	Indications on specific restrictions related to the type of traffic to be sampled (e.g. only vehicles, only trucks etc..)

In COLOMBO, and in the traffic surveillance solutions described in this deliverable, virtual sensors provide message primitives and containers to dynamically notify V2X equipped vehicles on what and how to do traffic surveillance when approaching an intersection.

3.2.3 Data Quality Requirements for Traffic Monitoring

The traffic light system developed within project COLOMBO requires two types of information related to traffic: the average speed of the vehicles and their number.

Vehicles speed is required for the low-level policy selection and it is translated into *pheromone* values, which is an abstraction of the traffic conditions ([COLOMBO D2.3, 2014], Section 2.1.5). This average value is required for every incoming and outgoing lane.

The number of vehicles, sensed or inferred, is used by the chain selection algorithm ([COLOMBO D2.3, 2014], Section 3.1) to determine which directions should be given green light. Also, this value affects the execution of two low-level policies, namely *Phase* and *Platoon* ([COLOMBO D2.3, 2014], Sections 2.1.1 and 2.1.2). This information is needed only for the incoming lanes.

The traffic light system was developed and tested with information for each individual lane. This kind of granularity might be too hard to achieve and data can be provided only for the whole roadway. This will not affect the behaviour of the traffic light system since speed is averaged among the lanes. The numbers of vehicles at different lanes that belong to the same roadway are considered together. Still, it may be interesting to differentiate data related to vehicles headed to different outgoing lanes, if it is meaningful (i.e. protected left turns [COLOMBO D2.2, 2014]).

The system proved to be robust to the different penetration rates of equipped vehicles, therefore robust to the accuracy of the information, provided that a proper tuning is available [COLOMBO D2.3, 2014].

4 V2X-based Traffic State Estimation

COLOMBO's DFCD aims at benefiting from the dedicated communication capabilities of V2X-equipped vehicles to extract and locally consolidate the required traffic data by COLOMBO's traffic light control.

V2X-equipped vehicles supporting the ETSI ITS communication system can cooperatively exchange their position/speed information. Instead of transmitting this to the cloud for traffic data consolidation, COLOMBO aims at consolidating traffic data locally either directly by vehicles, or by local RSUs.

A full penetration rate of V2X-equipped vehicles will be assumed at the first phase in order to validate the feasibility of the different approaches. In the second phase, a reduced penetration will illustrate the divergence between true traffic flows and the COLOMBO's estimated ones. This will build a bridge to the future COLOMBO deliverable D1.3, where alternate communication options (smart-phones, sensors) are tested for compensating the low penetration of V2X-equipped vehicles, providing traffic surveillance data as close as possible to the ground truth.

In this section, we present four different algorithms developed for the COLOMBO traffic surveillance.

- The first approach lets vehicles cooperate and locally estimate speed and flow in their neighbourhood before sending it to the RSU.
- The second approach takes a different direction and will estimate traffic flows from the relationship between multi-hop dissemination and traffic density.
- The last two approaches rely on the RSU to coordinate traffic surveillance by tracking their position and speed to estimate traffic flows.

4.1 Algorithm 1: Cluster-based Traffic Surveillance

A first family of protocol solutions and algorithms was developed that exploit the high-level guideline of grouping/clustering a set of neighbour vehicles, as detailed below. The primary reason for grouping/clustering is to maintain efficiency and scalability in high-density urban environments. The ultimate purpose of this set of V2V solutions is to deliver a comprehensive (yet approximate) stream of information regarding a wide set of properties of the traffic flow to the RSUs hosted at each traffic light. Although such information could be obtained by using a simple direct communication from vehicles to RSUs as described in Section 4.4 (Algorithm 3), scenarios with high densities of vehicles would lead this approach to generate a constant transmission of potentially high volumes of raw data (especially if many indicators are relevant for traffic management purposes). Accordingly, it could become difficult to effectively process/store and exchange data for currently available networking solutions.

In addition, as better described below, by introducing vehicle *groups* as an intermediate level of abstraction between the RSU and the multitude of surrounding vehicles, we can apply a data fusion process to keep the traffic flow information as simple and compact as possible and to filter-out unwanted data fluctuations. From here thereafter, a group is defined as a set of vehicles that are approaching a traffic light following a common direction. The central idea is that the leader node is chosen in an approximate central position, so that it can reach all other peers by simple single-hop communication and coordinate all group members. With a closer view on details, for each road that joins the intersection, the RSU stores the incoming direction (see Figure 4.1). The beacon sent by the RSU specifies the set of directions monitored by the corresponding traffic light: if the current average direction (considering a compliant error interval, as shown in Figure 4.1) of the receiving vehicle is conformant to any of these values, the node can start the group formation protocol, otherwise the packet is simply ignored (such as by vehicle c in Figure 4.1).

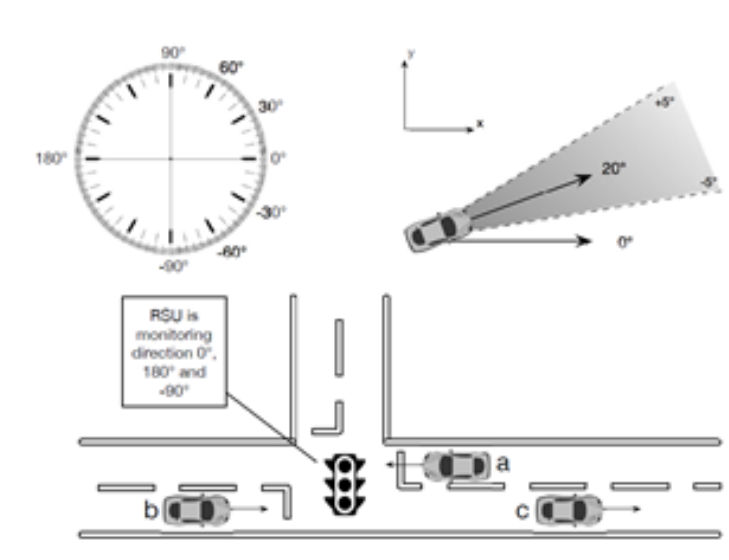


Figure 4.1: Definition of monitoring direction and direction compliance with an error margin.

In the following subsections, we briefly examine three main protocols. The first two are group formation protocols to create a new group. Two different formation protocols called *Reactive* and *Proactive Group Formation Protocols* (also in order to investigate and evaluate the suitability of differentiated approaches to group formation in COLOMBO-specific application scenarios) were developed. The main difference between them is that the proactive protocol has the prerequisite that nodes know information about their neighbours before the beginning of the group formation phase, while the reactive one has not. Once the group creation is over, the third *Group Management Protocol* is aimed to maintain the group active and to continuously collect information about group members until the group crosses the RSU; Group Management Protocol is independent of the protocol used for group formation.

4.1.1 Reactive Group Formation Protocol

Before starting with protocol description let us report some design guidelines we followed. First, a node can simultaneously be member of only one group. In addition, once inside a group, it continuously checks if its current direction is still conformant with the group direction.

The reactive group formation protocol consists of two phases: the first one to collect information about the neighbours of a node (see Figure 4.2-a, Figure 4.2-b, Figure 4.2-c, and Figure 4.2-d); the second one to elect the leader of the group (see Figure 4.2-e and Figure 4.2-f). In the first phase, the nodes are initially idle, listening for incoming messages. When a node receives a beacon message from an RSU, it checks if its current direction is conformant with the one contained in the received beacon (MT_GROUP_SETUP_REQ): if not, the node discards the message and stays idle; otherwise, it is directly activated by the message, sends a response message (MT_GROUP_SETUP_RESPONSE) and starts counting other response messages received from neighbour nodes for a randomly chosen timeout. An idle node that receives a response message can be indirectly activated if its direction is conformant with the one specified by the message. To avoid message flooding there is a maximum number of hops for indirect activation by a response message. The first node that ends its timeout sends a bid message (MT_GROUP_SETUP_BID) that ends the previous initial bootstrap phase and starts the leader election phase: the message contains the number of response messages received by the node. A node receiving a bid message stops listening for response messages even if its timeout is not elapsed, and raises the bid only if its value is above the current one. After a node has sent a bid, it waits for another randomly chosen timeout: when that timeout expires, if its bid is the highest, the node sends an IAmLeader message (MT_GROUP_SETUP_IAMLEADER) to inform that it is the group leader.

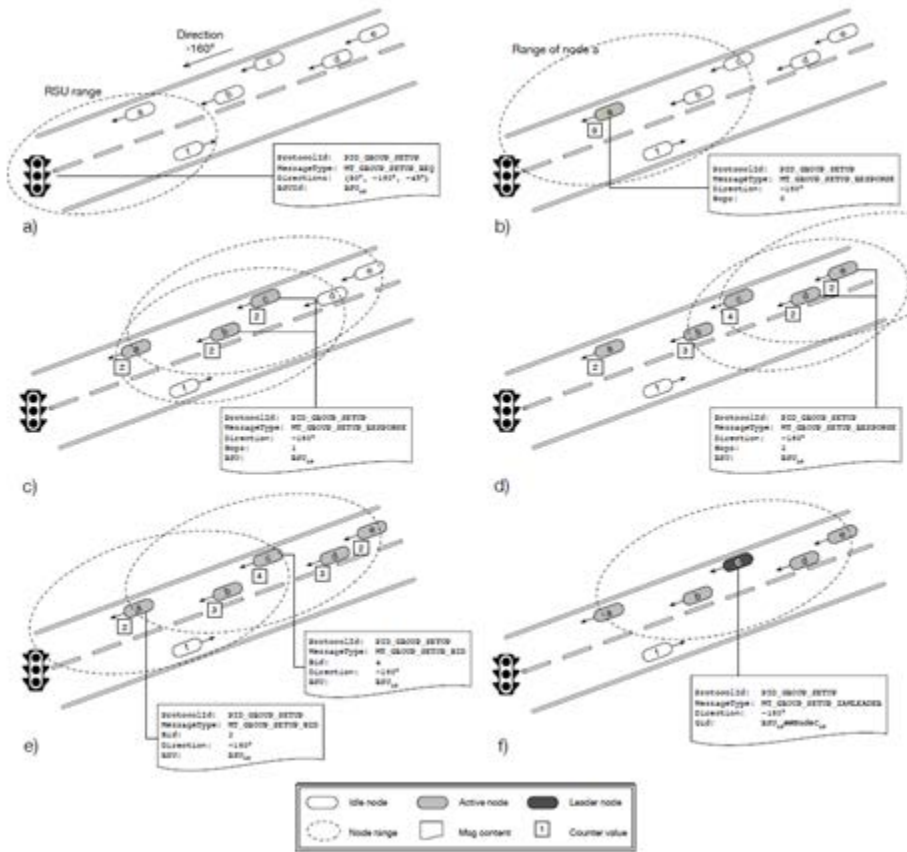


Figure 4.2: Group formation process with the reactive protocol. The RSU sends periodic beacon messages (a).

Every activated node sends a response message to its neighbours (b). Other nodes that receive the response message activate themselves and send a response if the maximum number of hops it has not been reached yet (c,d). The leader election phase is an auction that selects the best node to be leader (e). The node with the best bid is elected as leader of the group (f).

4.1.2 Proactive Group Formation Protocol

This protocol requires that the nodes have previous knowledge about their neighbours. Under this precondition, leader election is a simple operation because every node, when activated, has the needed knowledge to decide whether it is the best candidate to become leader (see Figure 4.3). If so, the node can declare itself as leader after a short timeout, otherwise, if the node knows a better candidate, it can send an invite message (MT_GROUP_SETUP_INVITE). When a node receives an invite message it checks if its direction is conformant, then, after a brief timeout, it declares itself as leader. A node stops the timeout if it receives an IAmLeader message from another vehicle. Timeout duration is selected dynamically to promote nodes with the highest number of known neighbours. A node with a large number of neighbours will wait less than one with a smaller number, so it is more likely to be the first to declare itself as leader.

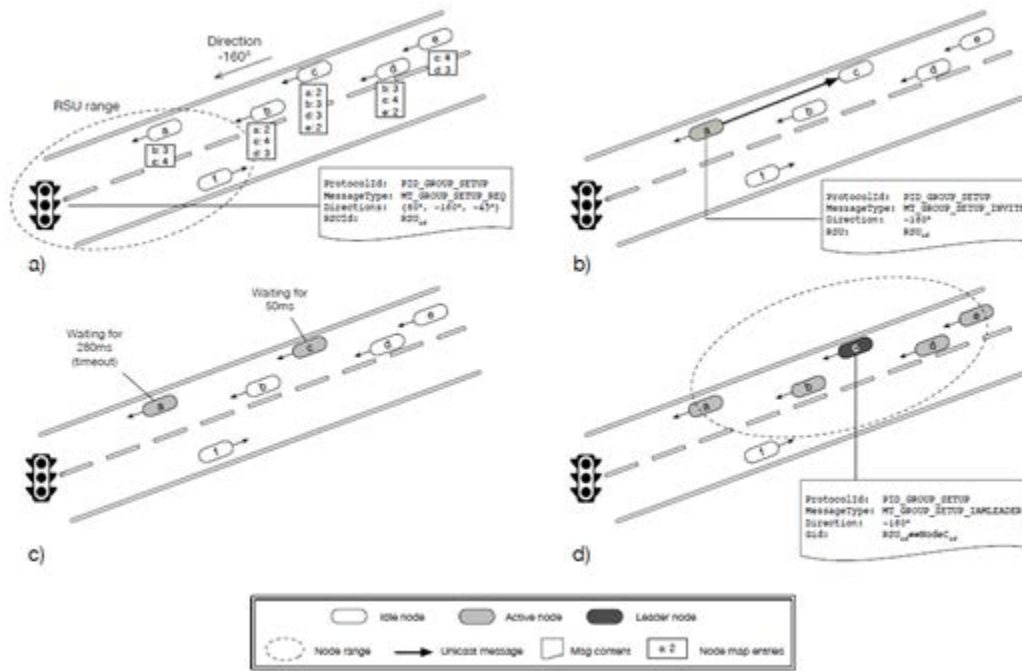


Figure 4.3: Group formation process with the proactive protocol. The RSU sends periodic beacon messages (a). Every node activated checks its node map and sends an invite message only to the selected node, if found (b). The nodes wait for a timeout inversely proportional to the number of known nodes (c). The node with the shortest timeout declares itself as leader (d).

To obtain previous knowledge about neighbouring nodes, even if in an idle state, they periodically send a heartbeat message, so that every node can keep a node map. The quality of the information stored in this map is proportional to the frequency of heartbeat messages.

4.1.3 Group Lifecycle Management

During the whole group lifecycle, the leader is responsible for harvesting and processing the available sensor information from all the group members as well as determining when the group can be safely disposed. To keep track of the status of its group, each leader keeps an internal node map with the most recent set of information available for its members (all locally sensed indicators of potential interest), such as position, speed, and direction. To continuously update this map, the leader periodically issues a probe procedure by sending a heartbeat message (MT_GROUP_BEACON) toward the group participants: all members must answer to this request within a given interval or the procedure will fail for that member. After a given number of consecutive failures, a member is considered no more available and removed from the group.

A member response packet also piggybacks all its available sensor information. After a single probe is completed, the leader updates the group-fused data (note that more complex and sophisticated algorithms for data fusion can be easily accommodated with the proposed solution) and sends it to the corresponding RSU – if reachable. In particular, the leader elaborates the information about the members of the group and sends the report with an info message (MT_GROUP_INFO). With this information the RSU can monitor the traffic evolution over time; in other words, once the group leader is in visibility of the RSU, it sends to the RSU multiple info messages to reinforce traffic surveillance data about the monitored group. This way, the traffic light can receive a higher-level stream of data that is already differentiated by the directions monitored for that intersection. The RSU is then free to further process these incoming information streams before taking any decision that will influence the traffic control algorithms.

If a node senses that its direction is no longer conformant to the one of the group, it simply stops replying to the leader heartbeats and reverts to a silent state. Similarly, an idle vehicle can overhear and inspect a leader probe packet and, if conformant to the group direction, answer to it to join that specific group. Finally, a group is disposed when its position crosses a selectable distance from the RSU (the default is 20 m). The position of a group is approximated by the position of its leader, since it was selected to be near the group centre. Additionally, a single member is allowed to autonomously leave a group if it senses that the distance between itself and the RSU is increasing. At the end of a probe phase, if the leader is inside the “disposal distance”, it sends a special info message to the group members and the RSU to inform them of group life termination. These actions are critical not only to ensure that vehicles that are leaving the intersection are excluded from data provisioning, but also to avoid excessive network traffic (and consequent packet collisions) in those areas that are more prone to suffer from higher node densities.

Note that the above protocols could be easily extended to consider not only group formation at one single hop from the targeted RSU, but also at n hops from there by simply modifying the initial propagation of the trigger message for group formation. This could be of high interest for all data collection and fusion operations requesting more time to be performed and related to monitoring data collected at a given (possibly large) distance from the collecting RSU.

4.1.4 Implementation in iCS

Our original protocols described above have been implemented on the iCS framework to enable a dynamic evolution of the traffic light control/management logics with the information collected by our V2X protocols. Every interaction between a client application and the core framework is obtained through a specific subscription. In addition, also to help/stimulate developers used to program ns-3 simulation code in passing to the rich iCS ecosystem, we have developed an adapter with the primary goal of increasing the level of abstraction, separating all the iCS-specific interactions, and offering a simpler interface for traditional simulation developers, as depicted in Figure 4.4.

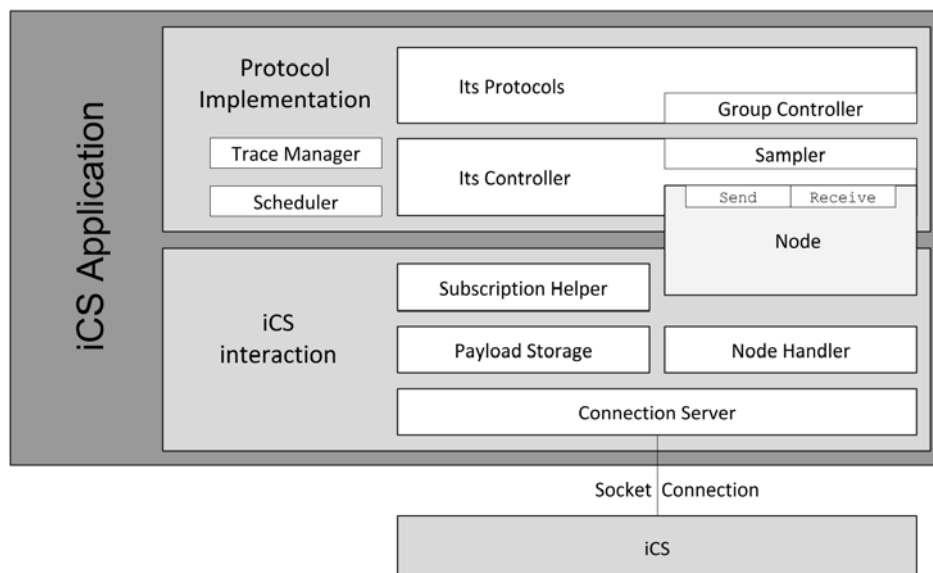


Figure 4.4: Architecture of the iCS application. The implementation of the protocol is on top of the adapter.

The iCS application is organized in two levels. The lower level manages all the interactions with the iCS framework and offers to the higher level APIs to send and receive messages similar to the one in ns-3. To that purpose, the lower level realizes several basic functions. A connection server handles the communications with the framework and interacts with the node handler whose task is to manage nodes. It handles nodes' lifecycle and manages the messages to and from the framework.

The node abstraction keeps track of node subscriptions and offers “usual” (traditionally exploited) communication primitives, namely, *Send* and *Receive*, thus making more immediate and facilitating any protocol implementation. This layer also contains a payload storage that allows the application to send only the identifier of a payload maintained with the framework, thus making the simulation execution more efficient and lightweight. The higher level contains the implementation of our protocols and of some of the support functionality of the ns-3 framework needed by them, such as a simple event scheduler, the same random generator used in ns-3 to improve commonality, and a simple tracing system.

In addition, to improve protocol simulation performance, several changes of the iCS framework were necessary. The most relevant ones relate to i) the reduction of the smallest simulation step from one second to one millisecond, ii) the alteration of the subscription to send and receive message to enable a geobroadcast mode of communication and the exchange of data between messages, and iii) changes to the interface between iCS and ns-3 to enable the scheduling of messages at a precise given instant instead of the beginning of a simulation step.

4.1.5 Simulation Results

All the V2X protocols above were prototyped and evaluated using the iCS, by comparing this implementation with an alternate one based on the usage of ns-3 alone. In our tests we used version 0.1.0 of iTETRIS, which integrates ns-3 version 3.7.0 and SUMO version 0.21.0; as for the standalone version of ns-3, we used version 3.18. All sensor information (e.g., position, speed, and direction) is managed by a low-level sampler component (see Figure 4.4). Such sampler component is responsible for introducing simple and stochastic error perturbations to both position and direction data as described in Appendix B.1. With this sampler component, it is possible to better mimic real-world sensor inaccuracies and to optionally differentiate classes of vehicles, equipped with different device/communication capabilities.

Moreover, to emulate realistic conditions with localization affected by errors, for each test presented herein, we introduce a position sampling error, modelled according to a normal distribution with mean=0 and a variance of 1.8 bounded to a 20 m maximum. At the network level, all communications set on the general iCS configuration are mapped and performed by the ns-3 standard YANS WiFi model using IEEE 802.11b in ad-hoc mode and a 1 Mbps bandwidth rate. The default log-distance propagation model is used to compute signal loss. Table 4.1 concisely reports the main configurations and parameters used in our simulations.

Table 4.1: Communication parameters configuration for iTETRIS experiments, those parameters are used for ns-3 configuration.

Parameter	Value
Wifi mode	802.11b (Ad-hoc)
Transmission mode	1 Mbps (DSSS)
Node radius	170 m
QoS	None
Propagation loss	Logarithmic
Propagation speed	Constant ($3 \cdot 10^8$ m/s)

By focusing on the deployment environments used to acquire experimental results, here we report results for the Pasubio multi-lane street intersection with a maximum of 4 lanes. In particular, we deeply evaluated and validated our solution under challenging peak hour traffic conditions: all simulations were performed in the same 30-minute daily timespan, during heavy traffic conditions experienced from 8:30 am to 9:00 am; during this time interval, vehicle density in the 170 m radius from the RSU can reach an average of 100 (130 as the maximum value). Vehicle densities change

during time according to a wave trend that follows the green and red timings controlled by the traffic light. Since pseudo-random distribution is frequently used by both the sampler component and the protocol logic, a batch of multiple independent runs with the same configuration but different seeds was used for every simulation setup.

Simulation Results

Although the proposed solution can harvest a much deeper range of traffic state properties, one central information for our traffic light control solution, according to the preliminary results obtained from WP2 is the quantity of vehicles approaching the RSU from a given direction. To effectively measure the performance of the proposed protocols, we compare the number of nodes estimated by *our solutions* with the number of vehicles sampled via a *real-mode* procedure, i.e., detecting (in god mode) the real position of every node in a 170 m range from the inspected RSU with full accuracy and precision. The value of 170 m is chosen to match a reasonable value for the maximum communication range of a mobile node.

Group formation protocols

We have collected a wide set of experimental results, which are published in [Bellavista, Foschini, Zagmani, 2014] and [Bellavista, Caselli, Foschini, 2014]. They relate to the performance of reactive and proactive group formation, with the participation of different mixes of Class B and Class C vehicles (associated with different precision/accuracy of positioning/speed information and with different communication capabilities, e.g., in terms of coverage range).

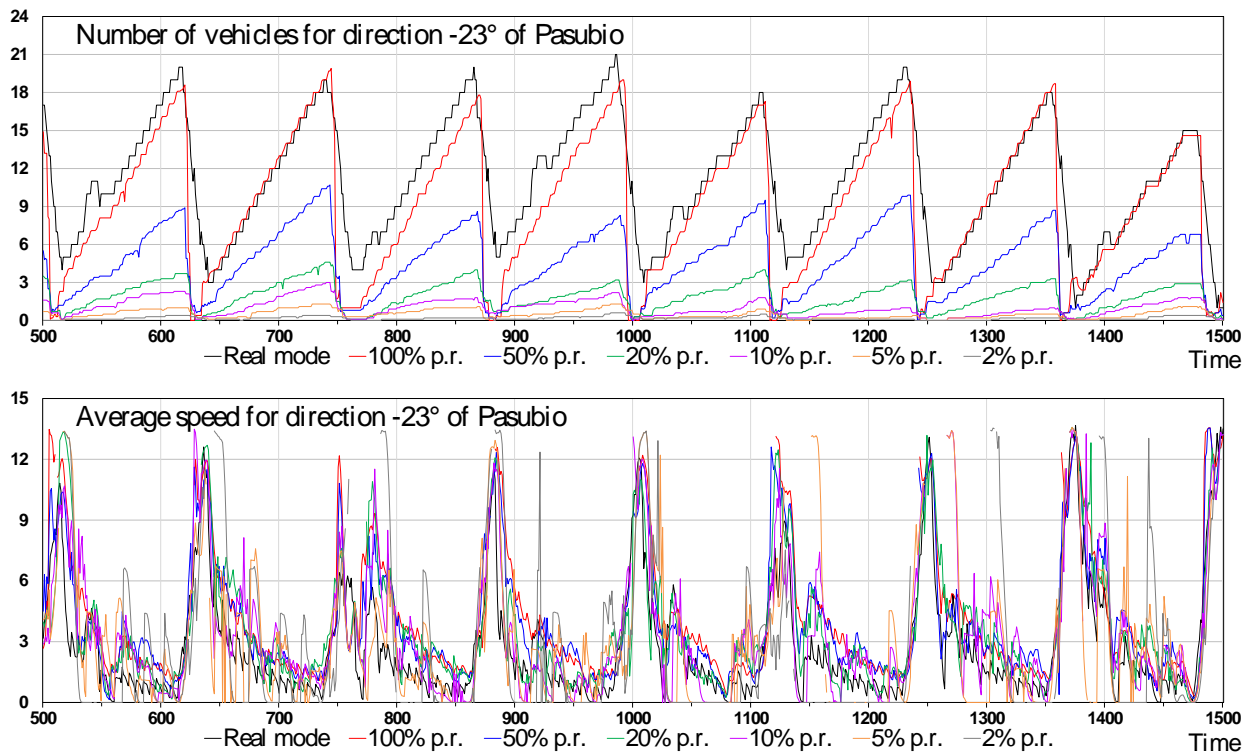


Figure 4.5: Results for different penetration rates.

In addition, other relevant experimental results that we collected refer to the investigation of our protocol behaviours at different penetration rates of the COLOMBO solutions. As a notable example, we report some of these results in the following. One central result of high relevance is

that the performance of our protocols has the very positive behaviour of decreasing linearly with the lower number of actively cooperating vehicles up to about a penetration ratio (p.r.) of 20 % for the number of vehicles. Below this threshold the protocol can exchange data only when a large number of vehicles are near the RSU, and this motivates the rapid decrease of the data quality. This fair behaviour enables linear corrections of the estimated vehicle density whenever one has coarse-grained information about average penetration rate or its approximated distribution over time/space. At the same time, focusing on velocity measurements, they are less dependent on the penetration rate; in fact, as shown in the bottom graph in Figure 4.6 following closely the real behaviour up to 5% p.r., and having reasonably good results even for the challenging 2 % p.r.

4.2 Algorithm 2: DissFlow Traffic Surveillance

Data dissemination in a vehicular network is critical to spread information related to safety or mobility. Vehicles transmit packets in unicast or broadcast to other vehicles in the dissemination direction. In case no vehicle would be in the transmit range, vehicles may also keep packets and wait until they meet further vehicles. As depicted on Figure 4.6, data dissemination may take two modes: *multi-hop* or *relay*, as function of the traffic density experienced by each vehicle. Estimating the dissemination delay for messages to reach a given area is of paramount importance for cooperative ITS applications, in particular traffic safety. Accordingly, several studies proposed analytical models, even integrated a penetration factor for V2X-equipped vehicles. All studies showed a strong link between the vehicular density on a street segment and the dissemination delay along that street segment.

While these studies could estimate the required dissemination delay if the traffic density is known, we are interested in the opposite case: can we obtain an estimate of the traffic density if the dissemination delay is known?

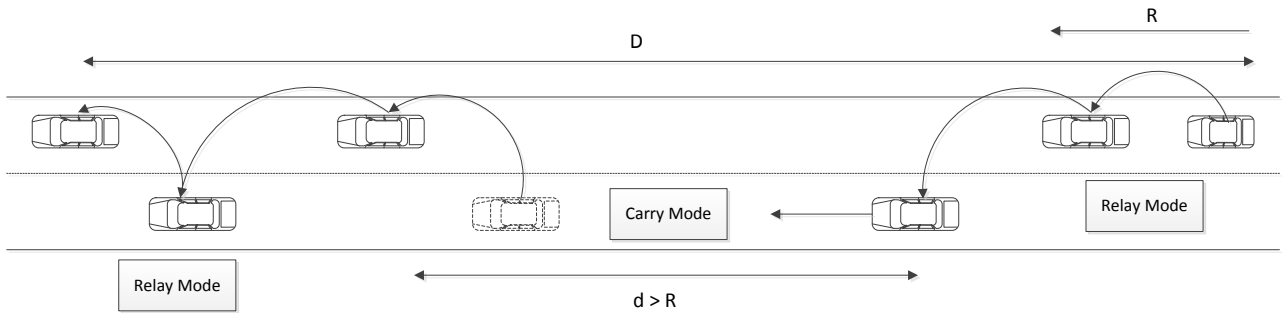


Figure 4.6: Dissemination mode as function of Vehicular Density, where ‘Relay Mode’ means vehicles relay packets, and ‘Carry Mode’ means vehicles keep packets.

4.2.1 System Description

In this work, we therefore propose a protocol and methodology to disseminate data between known locations, to record the delay and accordingly estimate the underlying traffic density.

Schematically speaking, the approach originates from the virtual sensor concept proposed by COLOMBO to provide traffic surveillance to traffic lights. As depicted on Figure 4.7, the road segment approaching a COLOMBO traffic light is segmented into several virtual sensing zones following TLC requirements. Each zone is handled by one virtual sensor, which objective is to estimate the traffic density in its zone. Each vehicle entering a virtual sensing zone will send a message to the traffic light, which will estimate the dissemination delay within that zone.

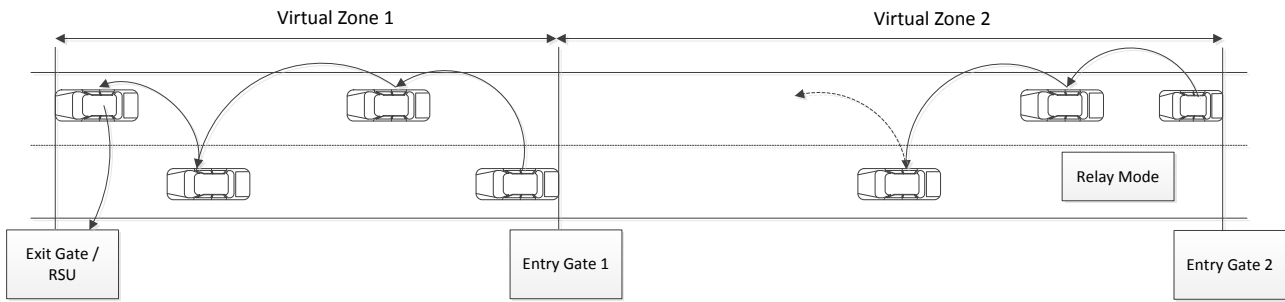


Figure 4.7: Configurable ‘Entry Gates’ as function of virtual sensing zones.

More specifically, a RSU, which can be directly attached to a COLOMBO traffic light control, or that is capable to connect to it, sends a TSM (Traffic Surveillance Management) message in broadcast with a transmit power capable of reaching all its virtual sensing zones. In the case some zones would be outside the range of the RSU, vehicles would be required to rebroadcast the TSM. A TSM message includes a set of GPS coordinates representing entry points of the virtual sensing zones. Each vehicle receiving a TSM message will compare its current location w.r.t the various entry points. Vehicles will not transmit a message if it already passed an entry point, but rather schedule to transmit a message when it will pass the next one.

Once a vehicle pass a given entry point, it will send a TSD (Traffic Surveillance Data) message back to the RSU. A TSD message includes the ID of the vehicle, a time stamp and its speed when the vehicle passed the entry point and sent the TSD message, and the current hop count. The dissemination protocol is based on the ETSI ITS GeoNetworking, which will let every vehicle providing the maximum progress relay the TSD message. If no such vehicle may be found, the TSD is carried by the vehicle until it finds such relay and further forwards it. To avoid wrong estimations of the directional traffic flow, only vehicles driving in the dissemination direction will be considered as relay. Upon each hop, a relaying vehicle will append to the TSD message its current speed, and increase the hop count by one.

Upon reception of a TSD message by the RSU, the RSU will extract the dissemination delay from the initial time stamp and the received time. It also can obtain the number of hops as well as the average speed along the road. The RSU extract an average dissemination delay, hop counts and average speed from all TSD messages received over a time frame TS_frame , and extract from a $TS_map(delay, hop, speed)$ function the corresponding traffic density.

As the traffic density can only be obtained between each entry points and the RSU, relative traffic density values corresponding to the virtual sensing zones are obtained by subtracting the respective inputs.

The proposed traffic surveillance approach is based on two key blocks: a **dissemination protocol** and a **mapping function**. Both are described below in larger details.

4.2.2 Dissemination Protocol

The traffic surveillance protocol is based on two messages: TSD and TSR, corresponding to the two phases of the protocol.

The first phase is to let vehicles know of the entry gates and when they should start sending a message back to the RSU. A TSD message (Traffic Surveillance Data) is therefore transmitted by RSU in geo-anycast and contains location of virtual zones and their respective entry gates. The composition of the TSD message is depicted in Table 4.2.

Table 4.2: Traffic Surveillance Management (TSM) Header.

Data	Data Requirement/Format	Optional/Required
RSU ID	MAC	Required
RSU Position	Geo-Coordinate	Required
Virtual Sensor Set	VIRTUAL_SENSOR type	Required
Record Speed	Boolean	Optional, true by default
Record Hops	Boolean	Optional, true by default
Revert Recording	Boolean	Optional, false by default

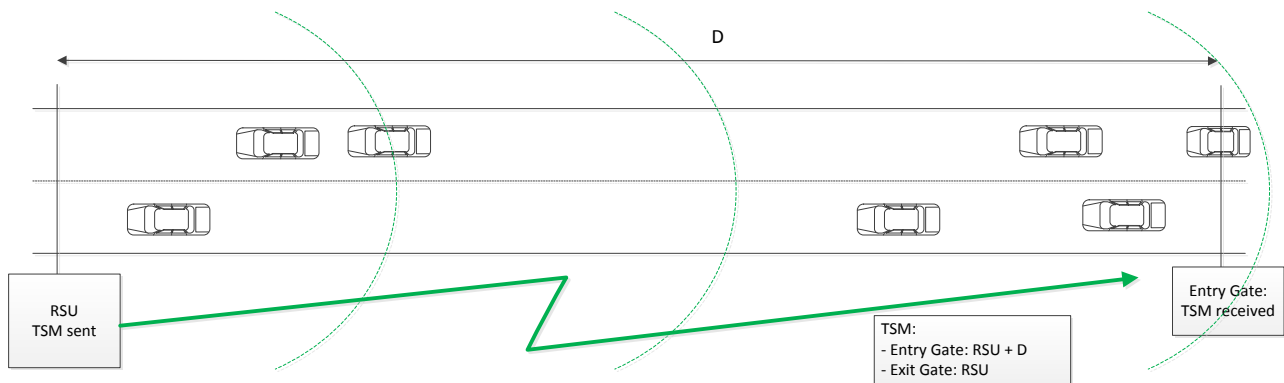
Where a VIRTUAL_SENSOR type is composed of the data shown in the following Table 4.3.

Table 4.3: Virtual Sensor Parameters.

Data	Data Requirement/Format	Optional/Required
Sensor ID	Unsigned Integer	Required
Entry Gate	Geo-Coordinate	Required

Table 4.4: Communication parameters configuration for iTETRIS ns-3.

Parameter	Value
Wifi mode	802.11a (Ad-hoc)
Transmission mode	6 Mbps (OFDM)
Node radius	170 m
QoS	None
Propagation loss	Logarithmic
Propagation speed	Constant ($3 \cdot 10^8$ m/s)

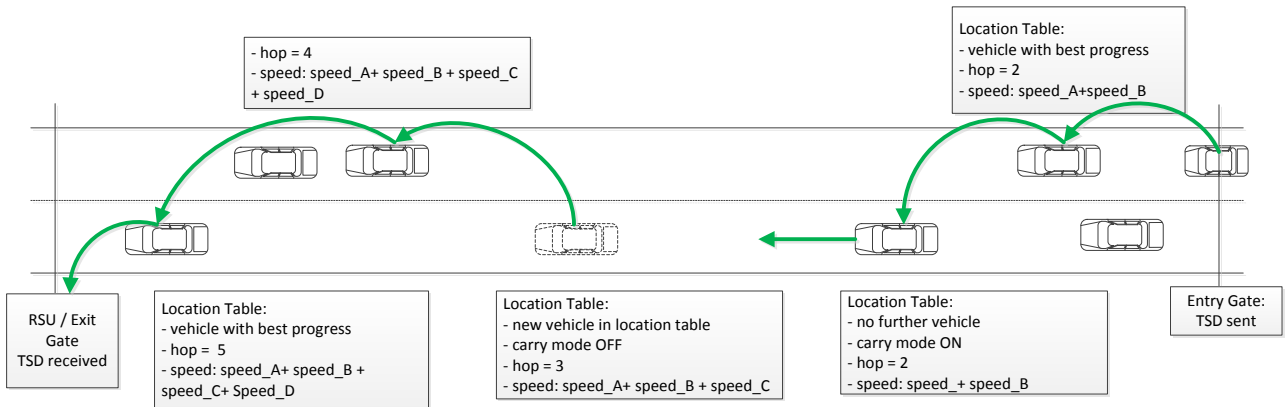
**Figure 4.8: RSU broadcasts the location of the entry gates of the virtual sensing zones.**

Accordingly, vehicles receiving a TSM message will only reply if they are at the entry gate of at least one of the virtual sensor zone indicated in the TSM message. Nodes that are within these zones will forward the packet, while any node outside of the zones will quietly discard the TSM message.

The second phase corresponds to the traffic surveillance per say. Upon the reception of a TSM message, a TSD message (Traffic Surveillance Data) is sent back to the RSU by any car crossing an entry gate in unicast. The TSD message is used to record the dissemination time, as well as other traffic information observed on the way. It contains data indicated in Table 4.5.

Table 4.5: Traffic Surveillance Data (TSD) Header.

Data	Data Requirement/Format	Optional/Required
Car ID	VIN/MAC	Required
Sensor ID	Unsigned Integer	Required
TX Time	Float	Required
Hop count	Short Integer	Optional
Speed Set	Vector of float	Optional
Revert Recording	Boolean	Optional, false by default

**Figure 4.9: Vehicles send a reply to RSU once passing through the gate.**

As illustrated on Figure 4.9, the initiator of a TSD message forwards the message to the next best relay bringing the maximum Euclidian progress towards the RSU. If no relay is found, the packet is stored and carried by the vehicle. Every vehicle relaying a TSD message will include its current speed and increase the hop count, so that the RSU will be able to get the number of relays and the average speed of these relays. Such data will be used to extrapolate traffic surveillance.

4.2.3 Dissemination Delay-Density Mapping Function

Two approaches are possible to obtain a delay-density mapping function. Either a function

$$\lambda = f(\Delta t) \quad (4-1)$$

can be used, where λ is the traffic density and Δt represents the dissemination time. Or, a function

$$\Delta t = f(\lambda) \quad (4-2)$$

can be used. For mathematical reasons, we opt for the latter and then propose an inversion formula to get $\lambda = f(\Delta t)$. An alternate approach would be to rely on a reverse table lookup function to extract λ from a known Δt .

The objective is therefore to obtain a continuous function capable of providing the dissemination delay given a traffic density. We define the problem as given in Figure 4.10. We divide a road segment in section of length R , where R represents the maximum transmission range. Within each section we assume two states: CONNECTED or DISCONNECTED. A connected zone represents a zone with sufficient vehicles to conduct a multi-hop relay, whereas a disconnected zone is a zone between two connected ones. We assume a message cannot be transmitted over a disconnected zone, which would require vehicles to carry the message.

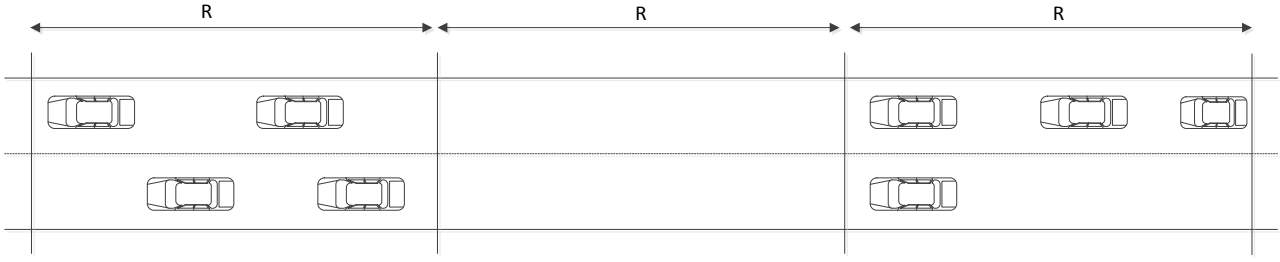


Figure 4.10: Connected and Disconnected Zones.

The analysis makes three key assumptions: first the inter-distance between vehicles is exponentially distributed. Second, the transmit range correspond to R , where any vehicles within the range may receive a packet with probability one.

According to the zone depicted in Figure 4.10, while a message is sent from an entry point to the RSU, it will experience CONNECTED and DISCONNECTED phases. We assume a cycle n consisting of exactly one CONNECTED and one DISCONNECTED phase. We also assume T_c^n and T_d^n the average time spent by the message in the connected, respectively disconnected phase at the n^{th} cycle. Without loss of generality we also assume that T_c^n and T_d^n are IID, and propose to model this process as an *alternative renewal process*.

Accordingly, $E[T_c] = E[T_c^n]$, and $E[T_d] = E[T_d^n]$, and we then define the probability of being in each of the two phases as:

$$P_c = \frac{E[T_c]}{E[T_c] + E[T_d]} \quad \text{and} \quad P_d = \frac{E[T_d]}{E[T_c] + E[T_d]} \quad (4-3)$$

We then define the Information Dissemination Speed (IDS) time as:

$$T_{IDS} = T_{hop} \cdot P_c + T_{carry} \cdot P_d = \frac{E[T_c] \cdot T_{hop} + E[T_d] \cdot T_{carry}}{E[T_c] + E[T_d]} \quad (4-4)$$

$$T_{IDS} = \frac{E[D_c] + E[D_d]}{E[D_c]/T_{hop} + E[D_d]/T_{carry}}$$

where $E[D_c]$ and $E[D_d]$ are the expected distance travelled while in connected and in disconnected phases, which we compute as follows.

While propagating, a message may encounter at least one vehicle, which means it is in a connected cell; or no vehicle at all, which means it is in a disconnected cell. We define $P_c = 1 - e^{-\lambda R}$ as the probability that at least one vehicle is found. Then, the number of successive cells passed before a disconnected cell is reached is defined as $E[N_c]$, and equivalently $E[N_d]$ as the number of successive disconnected cells. Conditioning on the first cell being connected, we obtain:

$$E[N_c] = (1 - E[N_c]) \cdot P_c + P_d = \frac{1}{P_d} \quad (4-5)$$

$$E[N_d] = (1 - E[N_d]) \cdot P_d + P_c = \frac{1}{P_c} \quad (4-6)$$

Accordingly,

$$E[D_c] = E[N_c] \cdot R = \frac{R}{P_d} \quad (4-7)$$

$$E[D_d] = E[N_d] \cdot R = \frac{R}{P_c} \quad (4-8)$$

Finally, substituting in T_{IDS} , we obtain the analytical model for the dissemination delay:

$$T_{IDS}(\lambda) = \frac{T_{hop} \cdot T_{carry}}{T_{carry} + P_d \cdot (T_{carry} - T_{hop})} = \frac{V_{hop} \cdot V_{vehicle}}{V_{vehicle} + P_d \cdot (V_{vehicle} - V_{hop})} \quad (4-9)$$

According to Eq. 4.9, the dissemination delay depends on the **vehicular speed** in the road segment, the **transmission speed** in a R distance and the probability of being in a disconnected mode $P_d = e^{-\lambda R}$.

Considering the following parameters: $V_{vehicle} = 20m/s$, $V_{hop} = 1000m/s$, $R=125m$, we can represent the dissemination delay as function of traffic density λ :

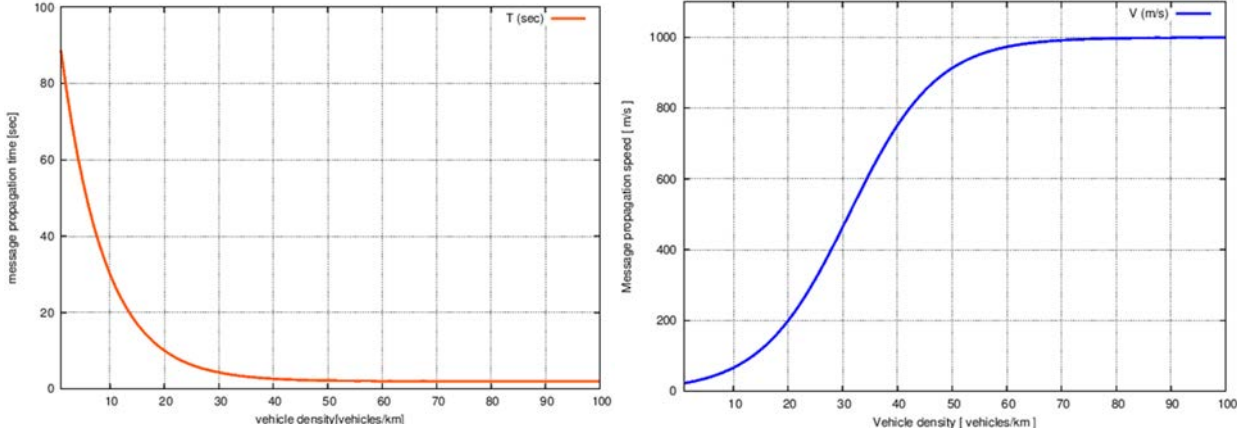


Figure 4.11: Dissemination delay (left) and Dissemination speed (right) as function of traffic density.

These curves represent analytical models given a particular road environment. The next step is to validate them using simulations on ns-3.

As the traffic surveillance application requires the traffic density as function of the dissemination speed, we reverse Eq. 4-9 and obtain:

$$\lambda(T_{IDS}) = -\frac{1}{R} \cdot \ln \left(\frac{V_{hop} \cdot V_{vehicle} - T_{IDS} \cdot V_{vehicle}}{T_{IDS} \cdot (V_{vehicle} - V_{hop})} \right) \quad (4-10)$$

Accordingly, Eq. 4.10 provides the traffic density given a dissemination time, which can be used by the traffic surveillance application to extract the current traffic density given the dissemination time of data packet transmitted either multi-hop or carried over the whole length of the road segment.

4.2.4 Implementation Details

The traffic surveillance application based on the dissemination delay / density relationship has been implemented on the application module, and also required small modifications to ns-3. ns-3 already includes a multi-hop geo-anycast implementation. Yet, when progress cannot be found, packets are discarded. For this traffic surveillance approach, a carry mode has to be implemented. Also, the geo-networking header has been enhanced to be able to add two fields: *mean_hop* and *mean_speed*, which are required by the traffic surveillance applications. Accordingly, the data dissemination algorithm works according to the state machine as illustrated on Figure 4.12.

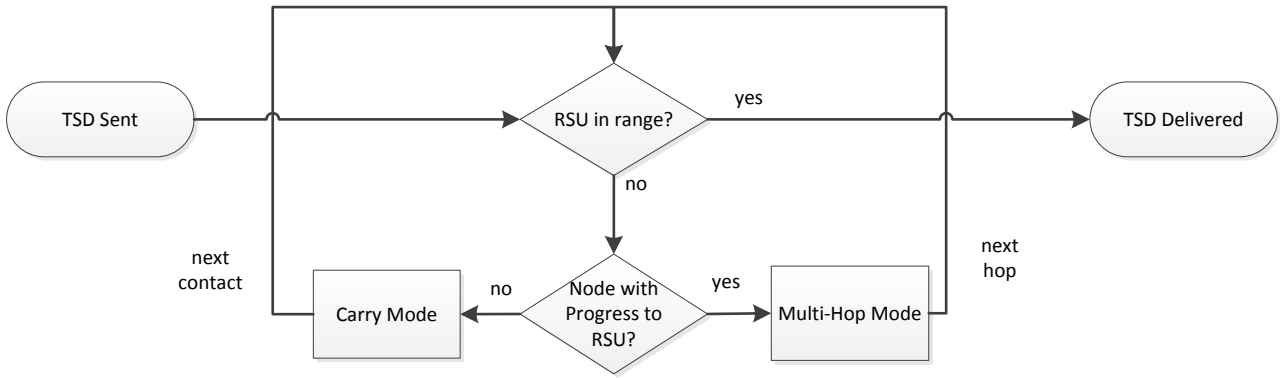


Figure 4.12: DissFlow Dissemination Strategy.

The application logics are only implemented on the application module. From an abstract perspective, it operates as indicated in Figure 4.13. The RSU sends periodically a TSR message that is received by all vehicles in the area. But only vehicles, which location corresponds to the entry zone of the virtual sensor will reply with a TSD message. The dissemination of the TSR is not monitored in this version of the traffic surveillance applications. The dissemination of the TSD sent back to the RSU is monitored. Upon reception of the TSD the RSU computes the delay between the transmission and reception of the TSD and extract the corresponding traffic density.

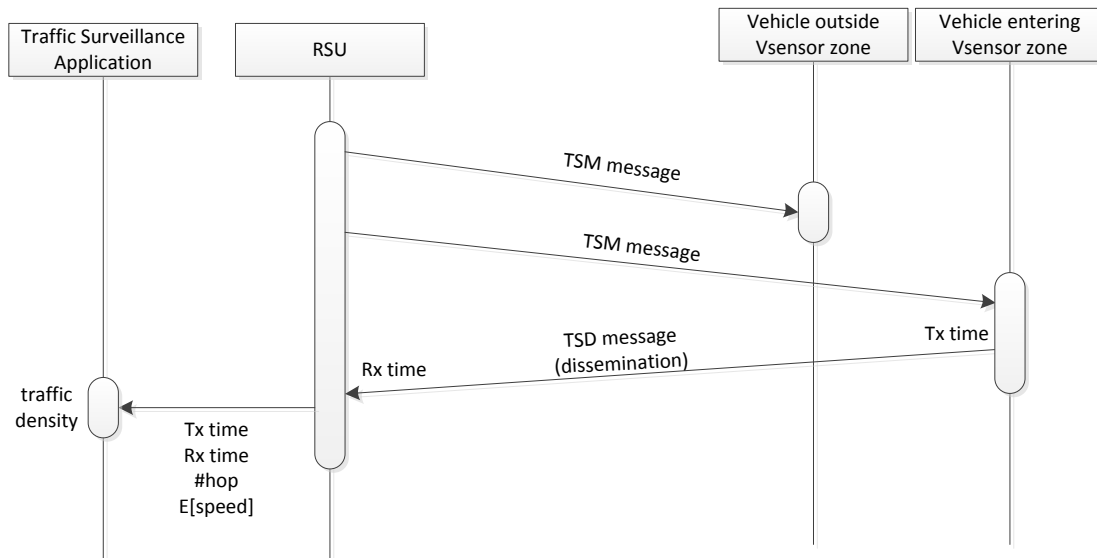


Figure 4.13: DissFlow Application Logic

4.2.5 Simulation Results

We provide in this section the proof-of-concept of this traffic surveillance approach. First, we validated the dissemination/density curves. For this purpose, we used the ns-3 module of iTETRIS as stand-alone. We configured a synthetic traffic scenario consisting on a 1000 m road reaching an intersection where a RSU is located. Vehicles passing the 1000 m distance from the RSU send one geo-unicast message to the RSU. The RSU consolidates the multi-hop dissemination delay for each given (and known) density. For each simulation, the density is therefore fixed. We consider the case of a 1-lane and a 2-lane road in order to see if higher flows on the second lane can impact dissemination.

Results are depicted on Figure 4.14 for single lane and on Figure 4.15 for two lanes. As it can be observed, the tendency of the simulation results follows the analytical one, although a drift is

clearly visible. This is due to two major aspects: first, the exact communication range cannot be adjusted as precisely as in an analytical model, so vehicles might receive TSD messages farther than the theoretical 125 m range. Second, the analytical model does not include the relaying and channel access delay, which increases exponentially with density. Both aspects will be integrated in the next phase. Nevertheless, we can see that beside the drift, the curve are capable of providing a dissemination delay given a traffic flow density.

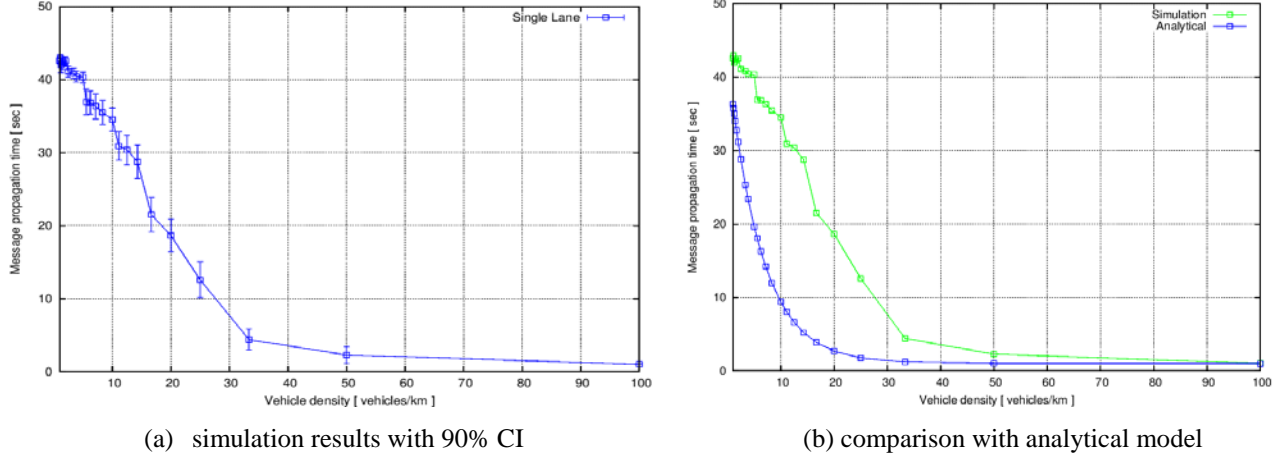


Figure 4.14: Dissemination delay vs. traffic density – 1 lane traffic.

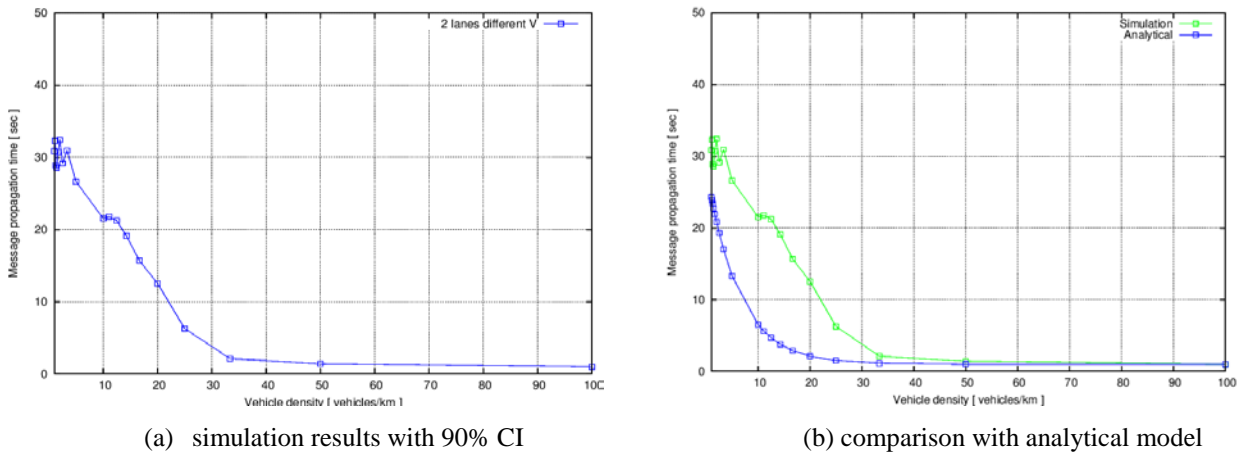


Figure 4.15: Dissemination delay vs. traffic density – 2-lanes traffic.

4.2.6 Summary and Outlook

As shown, the analytical curves showed to be close to more realistic traffic density vs. dissemination speed curves obtained by simulations. We also proposed an inversion formula to be able to obtain the traffic density as function of the dissemination speed $\lambda = f(\Delta t)$. These results prove that Eq. 4.10 may therefore be used to obtain a traffic density from a dissemination delay, including an adjusting factor. Eq. 4.10 will be the main function of the DissFlow protocol to provide traffic surveillance data to the traffic lights as function of cooperative V2X communications. The evaluation of the full DissFlow protocol is currently being conducted and will be provided in D1.3.

4.3 Algorithm 3: Traffic State Generation from CAMs

The basic idea of this approach is to exploit the information included in Cooperative Awareness Messages (CAMs), which are sent by V2X-equipped vehicles periodically. CAMs include a large number of different information that – given an RSU and equipped vehicles – can be collected at no additional hardware and communication bandwidth costs. In this section, a system that collects information from CAMs to compute performance indicators (PIs) is described. The system is

capable to compute a large number of different PIs, albeit with changing quality and ignoring environmental measures. The system relies on vehicles equipped with V2X technology and additionally needs an RSU that is located at the intersection to monitor. The RSU must be capable to receive and decode V2X CAM messages. Only this communication is required.

In the following, the proposed V2I-based traffic surveillance system is presented. At first, a coarse system description is given. The problem of mapping vehicles onto roads, necessary to obtain accordingly assigned PIs, is given afterwards. Then, the computation of PIs is described, showing which PIs can be computed. This is followed by evaluations of the system's performance in means of retrieving the correct traffic state at different penetration rates and for different aggregation intervals. The section closes with a summary and an outlook.

4.3.1 System Description

Vehicles send their CAMs (as usual) which are received by the monitoring RSU. As depicted in Figure 4.16, we assume one single RSU to be responsible for monitoring the traffic at a single intersection.

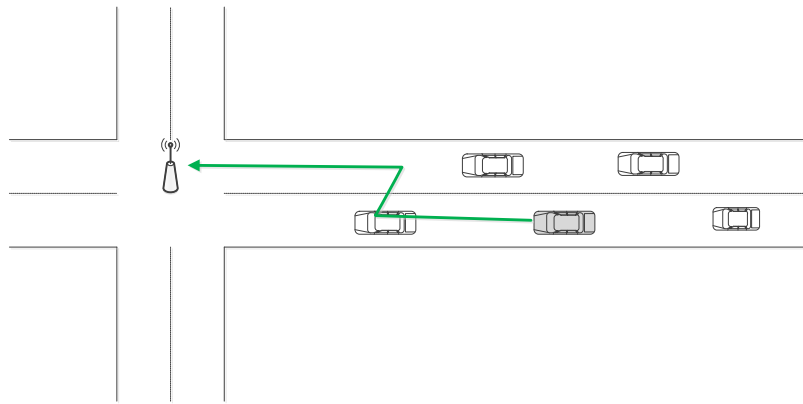


Figure 4.16: Sketch showing the communication participants (grey: equipped vehicle).

The system collects information from CAMs sent by vehicles within the communication range, which – if shadowing from buildings etc. is ignored – can be assumed to be a circle. But the PIs used in COLOMBO describe the situation in a road network or in front of an intersection, see [COLOMBO D1.1, 2014] (D1.1). Therefore, before computing the PIs, the position of the CAM sending vehicle has to be mapped onto the road network. After this assignment, the PIs can be computed. The overall procedure is depicted in Figure 4.17.

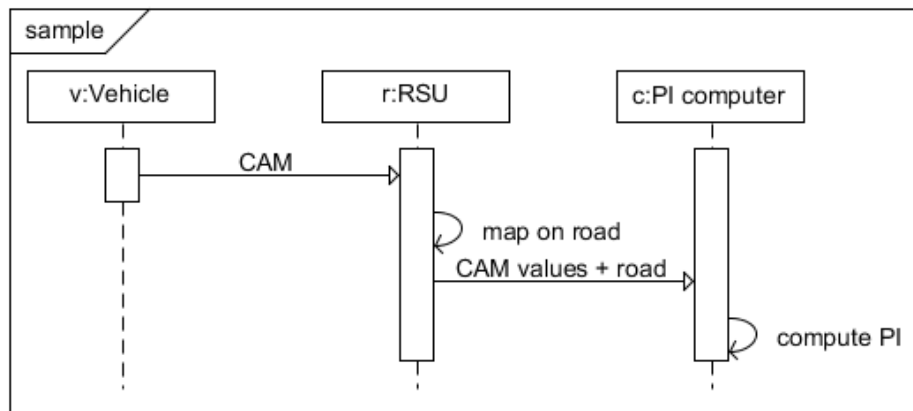


Figure 4.17: Process of PI computation.

4.3.2 Mapping Vehicles on Roads

It is assumed that PIs are only computed for roads and the place covered by intersections is not monitored. The major reason for this perspective is that the area within an intersection is shared by several streams and lanes are often only implicitly given. Thereby, it is complicated if not impossible to apply any kind of assignment of vehicle positions to roads and road positions within intersections. The second reason is that GPS errors should be taken into regard. Within urban areas, they are assumed to be about 10 m in diameter [Modsching et al., 2006]. Leaving out intersections introduces a margin that helps in determining the road the regarded vehicle uses.

The system maps vehicle positions on roads by simply determining which lane of the road network is closest to the given position of the vehicle and using the road this lane belongs to. Additionally, the angle is used to solve ambiguities on bi-directional roads. Figure 4.18 shows an example mapping where each dot represents a collected vehicle position. Red dots indicate positions that were not mapped due to being within intersections, green dots represent mapped positions.

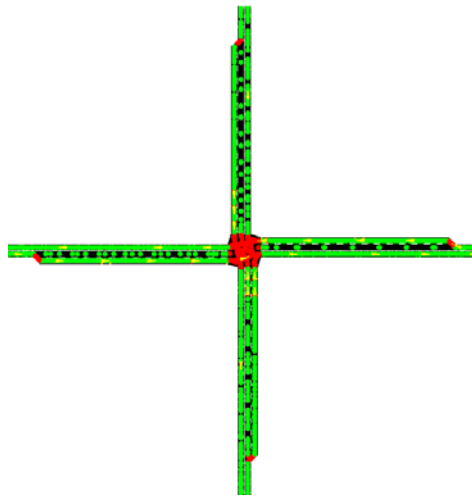


Figure 4.18: Examples of mapping vehicle positions onto the road network.

In the following evaluations, mapping errors are neglected to obtain a clear view on the remaining performance of the traffic surveillance system.

4.3.3 Performance Indicators Computation

As outlined, the system uses information sent by single vehicles that drive on known roads. In principle, only the speed information stored in CAMs is used to compute the PIs. Other information, such as the position of the vehicle or its driving direction is only used by the system to assign the vehicle to a road. Using the speed allows a trivial computation of the “*average speed*”, one of the performance indicators defined in D1.1. When introducing a threshold, it is as well possible to detect the halts of equipped vehicles, allowing to compute the PIs “*total waiting time*”, “*mean waiting time*”, and “*mean number of stops*”. As the system keeps the times at which a vehicle entered and left the communication range, the PIs “*total travel time*” and “*mean travel time*” can be computed, albeit only for the roads that are within the communication range. Having the average speed as well allows computing the delay time but this will be neglected in the following as the quality is equal to determining the speed itself.

Due to collecting all speed data individually (at 2 Hz - 10 Hz as defined for CAMs [ETSI-CAM, 2013]), the distributions of the basic measures (“*speed*”, “*waiting time*”, “*number of stops*”, and “*travel time*”) can be obtained, allowing to retrieve all relevant statistical properties, such as sum, average, median, standard deviation, percentiles, etc. In the following only the computation of PIs as defined in D1.1 is investigated.

Still, one further distinction should be named. The travel time can be only directly computed (it can be as well derived from the collected speeds) if a vehicle passes the communication range or an edge within it completely. The other named PIs can be computed for both, vehicles that are within the communication range, as well as only for vehicles that have left the range.

All the named computations regard only equipped vehicles. The system works under the assumption that some PIs may be directly used, even if only a low number of equipped vehicles is regarded. Average numbers are assumed to be directly corresponding to the according PIs when seeing all (equipped and unequipped) vehicles. Sums can be only computed by multiplying the obtained numbers with the inverse of the assumed penetration rate – which is known in the used simulation system, but which may be only assumed for real-world conditions. One hypothesis is that extrapolating measures by using an assumed equipment rate does not work well. These assumptions will be verified in subsequent sections.

4.3.4 Simulation Results

For evaluating the system, a simple scenario consisting of one intersection and flow from only one direction was chosen – neither infrastructure nor traffic light settings are assumed to influence the quality of the performed surveillance in the following. Due to being limited by the traffic light, the time-space patterns (such as the number of vehicles on the regarded road or their speed) are quite complex yielding in strong fluctuations of the performance indicators. The subsequent evaluations show the development of the PIs for one edge of this scenario, only.

Table 4.6 and Table 4.7 show the obtained PIs when regarding an aggregation interval of 300 s (5 min). The tables show respectively the behaviour for vehicles that have left the observation range (either the communication range or the observed edge). They include all respectively obtained measures. One obvious issue is the lack of data for intervals where no equipped vehicle was sensed. The probability to have no data for an interval depends on the aggregation interval's duration and the penetration rate. For this reason, low penetration rates show data lacks at times where no equipped vehicle has been within the communication range.

Table 4.6: PIs as obtained from vehicles that have left the area within the respective time interval.

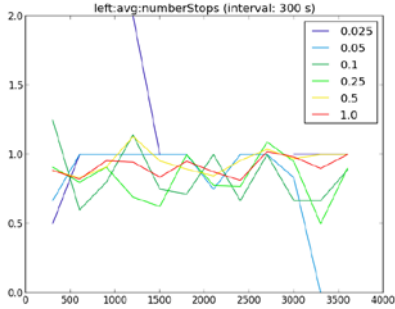
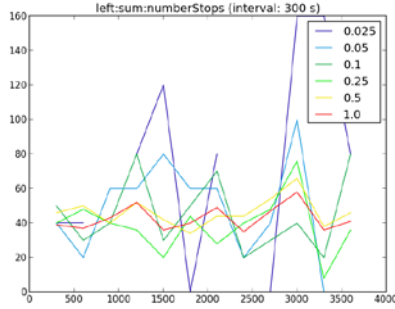
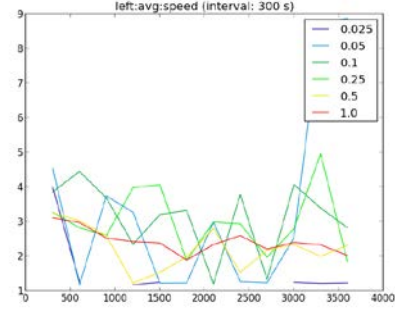
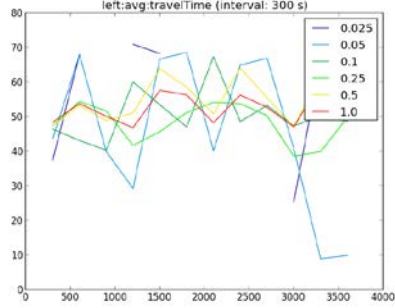
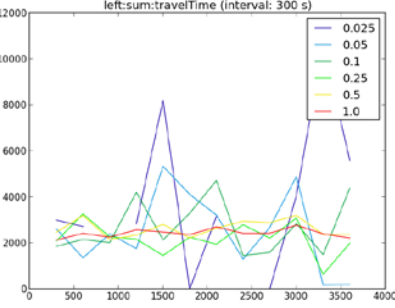
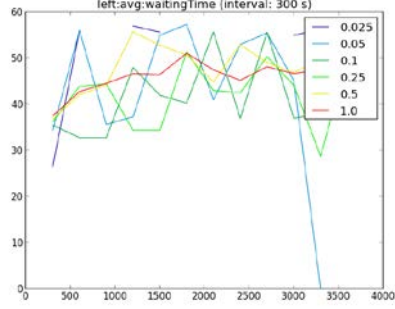
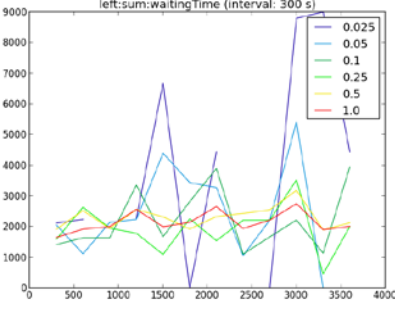
Measure	average	sum
number of stops		
speed		N/A
travel time		
waiting time		

Table 4.7: PIs as obtained from vehicles that were within the area within the respective time interval.

Measure	average	sum
number of stops		
speed		N/A
waiting time		

Besides the penetration rate, the results differ in dependence to the aggregation interval as well. The following tables Table 4.8 and Table 4.9 summarise the error in dependence to the aggregation time and the penetration rate, again respectively differing between methods that regard only vehicles that have left the observation range and all vehicles. For each interval and penetration rate, the error is computed as:

$$E(\text{rate}, \text{interval}) = (M(1, \text{interval}) - M(\text{rate}, \text{interval})) / M(1, \text{interval}) \quad (4-10)$$

Where:

rate: the penetration rate, in [%];

interval: the aggregation interval length, in [s];

$E(\text{rate}, \text{interval})$: the error for a given penetration rate and aggregation interval length;

$M(\text{rate}, \text{interval})$: the measure at the given penetration rate and aggregation interval length.

Table 4.8: Errors in dependence to the penetration rate and the aggregation interval (vehicles that have left the communication range).

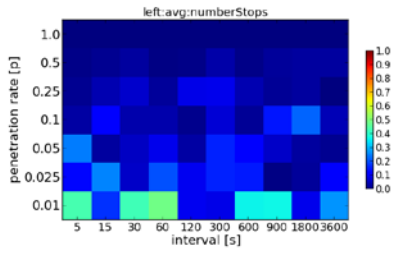
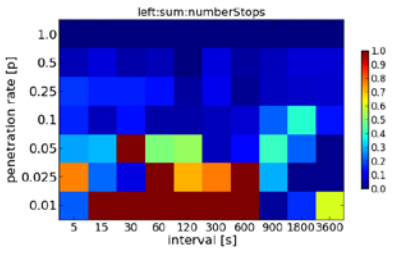
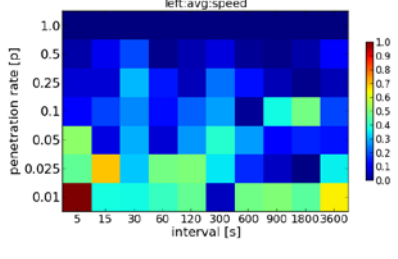
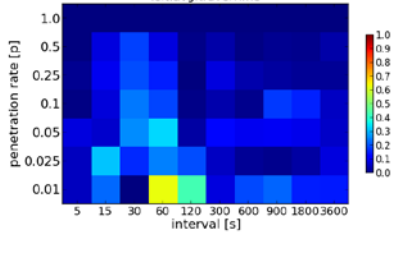
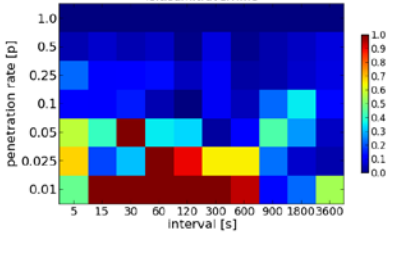
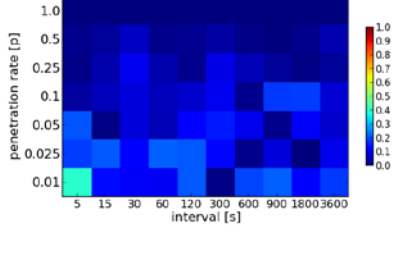
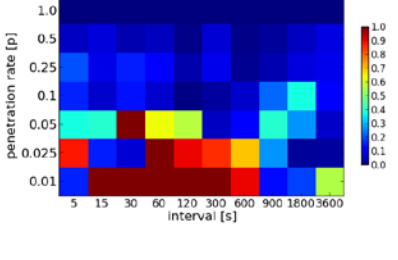
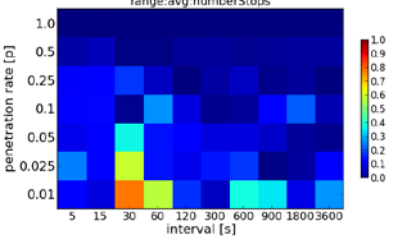
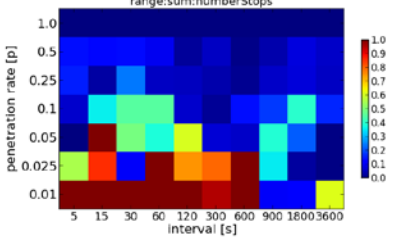
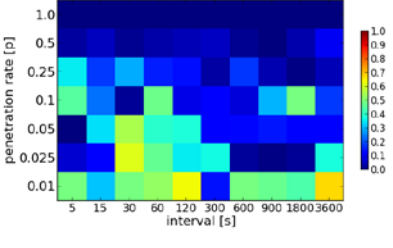
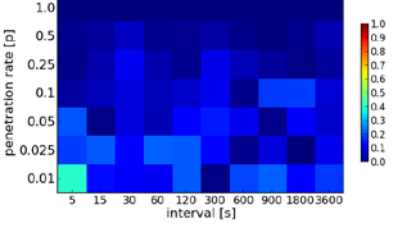
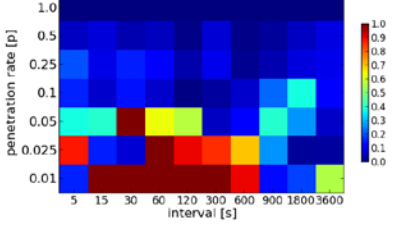
Measure	average	sum
number of stops		
speed		N/A
travel time		
waiting time		

Table 4.9: Errors in dependence to the penetration rate and the aggregation interval (vehicles that have left the communication range).

Measure	average	sum
number of stops		
speed		N/A
waiting time		

4.3.5 Summary and Outlook

As shown, already deployed V2X road side units can be a source of information about traffic efficiency. As assumed, the penetration rate is the major factor that influences the quality of the surveillance, but as well the aggregation interval to report should be taken into account.

When regarding a low number of equipped vehicles, one could coarsely assume that absolute numbers (like the sum of stops) can be hardly determined, while averaging measures (like the average velocity) can be retrieved with a sufficient quality. These assumptions were confirmed using simulations. It would be as well possible to distinguish different vehicle types as given in CAMs. Further investigations on this topic have not been performed, as the performance in means of correctly computing PIs for distinct classes should follow the evaluations of the overall fleet.

The presented results are based on a very simplified communication model that uses the distance only. As well, GPS errors have been neglected. Further investigations should be performed that use the complete COLOMBO simulation system to evaluate the performance under more realistic conditions.

4.4 Algorithm 4: Probe Vehicle Data-based Traffic Surveillance

Currently, many types of detectors are used at various locations on intersections. Their main functions and locations are summarized in Figure 4.19. Vehicle actuated controllers are based on presence detection. No vehicle present at the stop line means that a signal group can be skipped, while an occupied gap detector will extend a green phase. This also holds for the bicycle and pedestrian detectors, which are often push buttons for presence detection. For traffic adaptive control counting is important; the traffic light systems Utopia and Imflow have queuing models that use counts from entry, stop line and exit detectors. Not all three detector placements have to be present when upstream intersection data can be used what must be supported by the topology of the intersection and the network. Traffic responsive methods use counts of these detectors as well for selecting the right timing plan for the current demand pattern (c.f. [COLOMBO D2.2, 2014]).

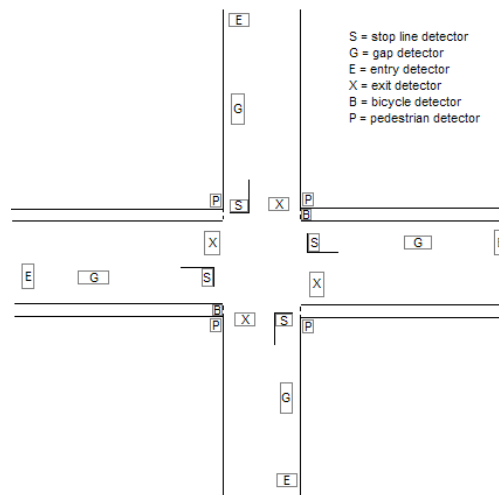


Figure 4.19: Common detector locations.

With probe vehicle data neither presence detection nor counting are directly possible. Reliable presence detection will only be possible with near 100 % penetration of cooperative vehicles and accurate queue counting would also require a significant and stable penetration. The basic principles of SCOOTs can be followed with low penetration probe vehicle data. Adjusting cycle time, offset and split time per signal group is possible based on the more slowly updated values of average speed in a road section. This is because speed gives an indication about queue length and green wave coordination.

When the algorithm controls an isolated intersection, coordination can be kept out of the equation. To have an idea if there is a queue or not, the travel time should be evaluated, which is composed of the free-flow travel time and the delay time at the intersection. Given that the signal group is exactly at saturation and has a uniform arrival pattern, the queue-time diagram looks like depicted in Figure 4.20.

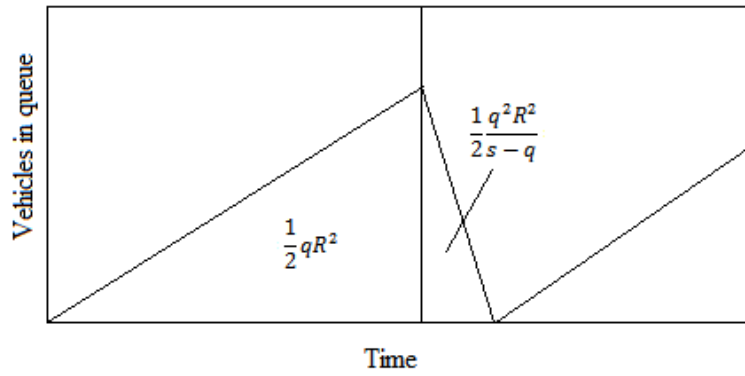


Figure 4.20: Queue-Time relation for uniform arrival pattern

The formula for total delay time (T_{tw}) is as follows:

$$T_{tw} = \frac{1}{2} q R^2 + \frac{1}{2} \frac{q^2 R^2}{s-q} = \frac{s q R^2}{2(s-q)} \quad (4-11)$$

In this formula q is the arrival rate, R the red time per cycle and s the saturation flow. The first term in this formula is the total delay time during red and the second during green. The surface in Figure 4.20 is made up of two triangles of which the highest point is $q.R$. The slope of the second triangle is $s-q$ and the duration is the height $q.R$ divided by the slope $s-q$. When the two terms are added the last simplification in the equation can be made.

To get to the average delay time per vehicle (T_{ad}) this result has to be divided by the amount of vehicles which experience this delay:

$$T_{ad} = \frac{T_{tw}}{qR + \frac{Rq^2}{(s-q)}} = \frac{s q R^2}{2(s-q) \left(1 + \frac{q}{s-q}\right) R q} = \frac{s R}{2(s-q+q)} = \frac{R}{2} \quad (4-12)$$

In the first step, the total delay time is divided by the number of vehicles arriving during red and the number of vehicles arriving during green while there is still a queue. The following steps are simplifications to finally arrive at the result $R/2$.

From this result it can be concluded that when a signal group is exactly at its saturation point, the average delay should be half of the red-time per cycle. When it is below saturation, the vehicles that queue up still have this same average delay time, but there will be a part of the vehicles that arrive during green while there is no queue. Therefore, the average delay time will drop below half of the red time. Above the saturation point, at least one vehicle will have to wait more than one cycle, so the average delay time will be higher than $R/2$. As long as the green time is too short for the demand, this average delay will keep rising. Once the delay time is very long, and enough capacity is allocated to the signal group, the buffered queue should first be processed before the travel time will go back to $R/2$ or lower.

So far this theory assumes a uniform arrival pattern, however in reality this is a Poisson distribution for isolated intersections. Therefore, averaging over multiple cycles will be necessary to deal with fluctuations between cycles. When another intersection is close enough to influence the arrival pattern, coordination has to be taken into account. In this case the arrival flow q is not constant and the set point for saturation will be at a different point than $R/2$. In case most vehicles arrive during the end of red/beginning of green, the average delay will be lower and when most arrive at the beginning of red it will be higher. Another problem arises when there are few vehicles on a certain signal group that report their travel time. In this case high fluctuations can occur due to the moment during the cycle at which these few vehicles arrived. This can be solved with averaging out over multiple cycles, but also by reporting the maximum and minimum travel times to the controller.

The key for measuring the traffic flow in practice is to assess whether the change in a measurement was caused by different coordination, different traffic light capacity or a change in external factors like average free flow speed or demand volume. When the situation is simple and free flow speed is constant while the intersection is isolated and the arrival pattern follows a Poisson distribution, the saturation rate is proportional to the average delay time as a percentage of the red time. When the average delay time is $0.5R$, the signal group is at its saturation point. At a lower value of $0.4R$, one fifth of all vehicles arrive during green, meaning that the traffic flow is only equal to the capacity of a green phase that is 20% of the cycle time shorter than the used green phase length. Above the saturation point, the average delay time can for example be $0.6R$. In that case 10 % of the vehicles had to wait an extra cycle. Those vehicles have the same delay time as the others plus one extra cycle. If the previous measure of the average delay time was still at $0.5R$ or lower, then this 10 % of vehicles were excess flow and the traffic flow is 10 % higher than the capacity of the green phase length.

For non-Poisson arrival processes, a correction factor should be received from the detection algorithm for a new saturation set point. This set point is specific per control type and can therefore not be described in a generic formula. With the new set point the same logic for determining the flow with respect to the length of the green phase applies. Another parameter received from the detection algorithm is the maximum travel time. This is to cope with low penetration information and low volume signal groups. These may only have a few detections per evaluation period, resulting in a value that has a lot of random noise and can easily fall between $0.5R$ and R without the presence of a queue that is longer than the capacity of one green phase. In this case the maximum travel time value contains valuable information. When the queue is too long to clear in one green phase, there should be at least one vehicle with a travel time higher than R . This method can also be applied when the arrival process is non-random but unknown. The same way the minimum travel time can also be used to prevent underestimation of the traffic flow.

The previous theory could directly be used for a control algorithm. The fluctuations of the measurements should be averaged out. This is to prevent too strong reactions to inaccurate measurements when there are just few vehicles reporting their travel time. However, when a queue is building up and the average delay time is above R , the traffic light control algorithm should respond quickly with increasing the green time.

5 Outlook at Further Surveillance Applications

The traffic surveillance methods presented so far aim at replacing conventional stationary detectors. Thereby, they deliver traffic efficiency measures similar to the latter. Within COLOMBO's WP1 two further applications are developed that shall deliver new, not yet existing measures. The first one is an incident monitoring system, capable to detect driving anomalies and hazards and thereby addressing traffic safety. The second one is a local emissions monitoring system, where "local" denotes the area of an intersection equipped with a V2X-capable road side unit.

When assuming an existing V2X-based traffic light, as developed in WP 2, both applications re-use the existing communication hardware and shall be thereby deployable at the cost of a software update. Both applications are scheduled to be presented in this WP's subsequent deliverable D1.3 (due in M30, April 2015). To show the current progress, the work performed so far is outlined in the following subsections.

5.1 Driving Anomalies and Hazard Detection

Anomalies can be detected at a macroscopic and at a microscopic scale. On the one hand, macroscopic changes in the traffic flow's attributes indicate an incident. Here, the solutions presented in Chapter 4 can be used. On the other hand, the fine-grained information retrieved from V2X-communication is capable to deliver trajectories with a very high resolution – single vehicle data at 2 Hz to 10 Hz. This invites to use them not only for determining aggregate traffic efficiency measures or performance indicators, but as well to exploit their applicability to microscopic (vehicle-oriented) safety-relevant questions. Thereby, to discuss what a driving anomalies detection can do and what not, the following aspects may be considered:

1. Incident detection: a road is partially or completely blocked and causes a traffic jam to develop. In this case, the speed of the vehicle(s) changes unexpectedly.
2. Small incident: a hindrance blocks part of the road, but in this case no traffic jam develops, the vehicle trajectories simply depart from their normal behaviour. It manifests in this case by a place on the road where vehicles normally drive, but not now. The acceleration behaviour is changed slightly, but not strong enough to get detected.
3. Driving anomaly: a vehicle behaves different than the majority of vehicles in the given situation, what may include bigger lateral accelerations, strong longitudinal decelerations, a too high speed and similar.
4. A critical situation: a vehicle has to brake very strongly to avoid an accident, or to change course abruptly to avoid another vehicle or any other moving hindrance. This can be detected either by the acceleration behaviour, or by some traffic safety measure indicating a dangerous situation.

All these cases cause the driving to become different from the normal case and are therefore subject to driving anomalies detection.

Approaches towards recognizing incidents or hazards on a microscopic scale are as well targeted by image-processing techniques. Here, video pictures taken by a camera located at an intersection are the source of vehicle positions. When processing a video stream in real-time, one obtains the vehicles' trajectories. Figure 5.1 shows the trajectories obtained using such systems. Video-data can in principle be as well used to test some of the ideas regarding the quality and reproducibility of safety measures, as done e.g. by the DLR in the past (see [Meysel, 2008], [Saul et al., 2014], [Jost et al., 2011]). The results can then be easily generalized and transferred to situations where the data do not stem from video, but instead from traces generated by vehicle-to-infrastructure communication.

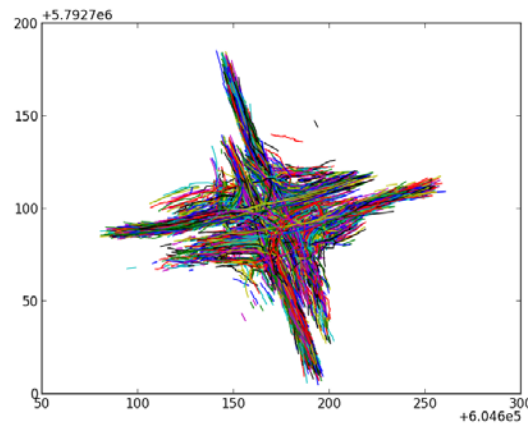


Figure 5.1: Vehicle trajectories obtained from image processing of a video stream.

Still, especially the microscopic approaches are currently assumed to be hardly functional at lower penetration rates. The probability of an incident is already very low³. By assuming quite deliberately a probability for critical or anomalous driving p_c of $1,5 \cdot 10^{-4}$ (1/s), an equipment rate $\eta = 0.01$, and a daily traffic volume of 1,000 vehicles, which are for 50 seconds in the spot under consideration, an observation time of 3 days is needed to record just 10 critical situations. Note, that this is as well true for the more favourable case where each vehicle can compute its own safety measure.

Of course, not all V2X-based applications related to traffic safety trigger an event that seldom, at least approaches addressing the macroscopic are estimated to be reasonable.

5.2 Local Emissions Monitoring

Within COLOMBO, vehicular emissions are covered by the Work Package 4, including the design, implementation, and validation of new emission models as well as investigations of relations between traffic light control settings and emissions and the design of emission-optimal driver behaviour (see also [COLOMBO 4.3, 2014]). Still, within COLOMBO's WP1, Task 1.5 is dedicated to the development of a system that – deployed at a traffic light's RSU – determines the amount of pollutants emitted by vehicles that cross this intersection. So far, the system has been designed and implemented and first steps towards measuring its performance in means of correctly computing the amount of emissions have been taken. The performance of the system, including its learning phase is shown in Figure 5.2.

The simulation results show that even with a low penetration rate of 1 % the relative error of the estimated emissions is around 5 % after 1 hour of learning time. The relative error is decreasing with higher penetration rates. It has to be mentioned that the used simulation scenario is very simple so with more complex traffic conditions the error will probably be higher.

³ Note, that a very rough estimate says that an accident occurs every 200,000 km driven on urban German roads. By assuming a speed of 14 m/s, this is one accident in $1,5 \times 10^7$ s travel time. Critical situations are at least 1,000 times higher, so the number assumed above for p_c is not completely unrealistic.

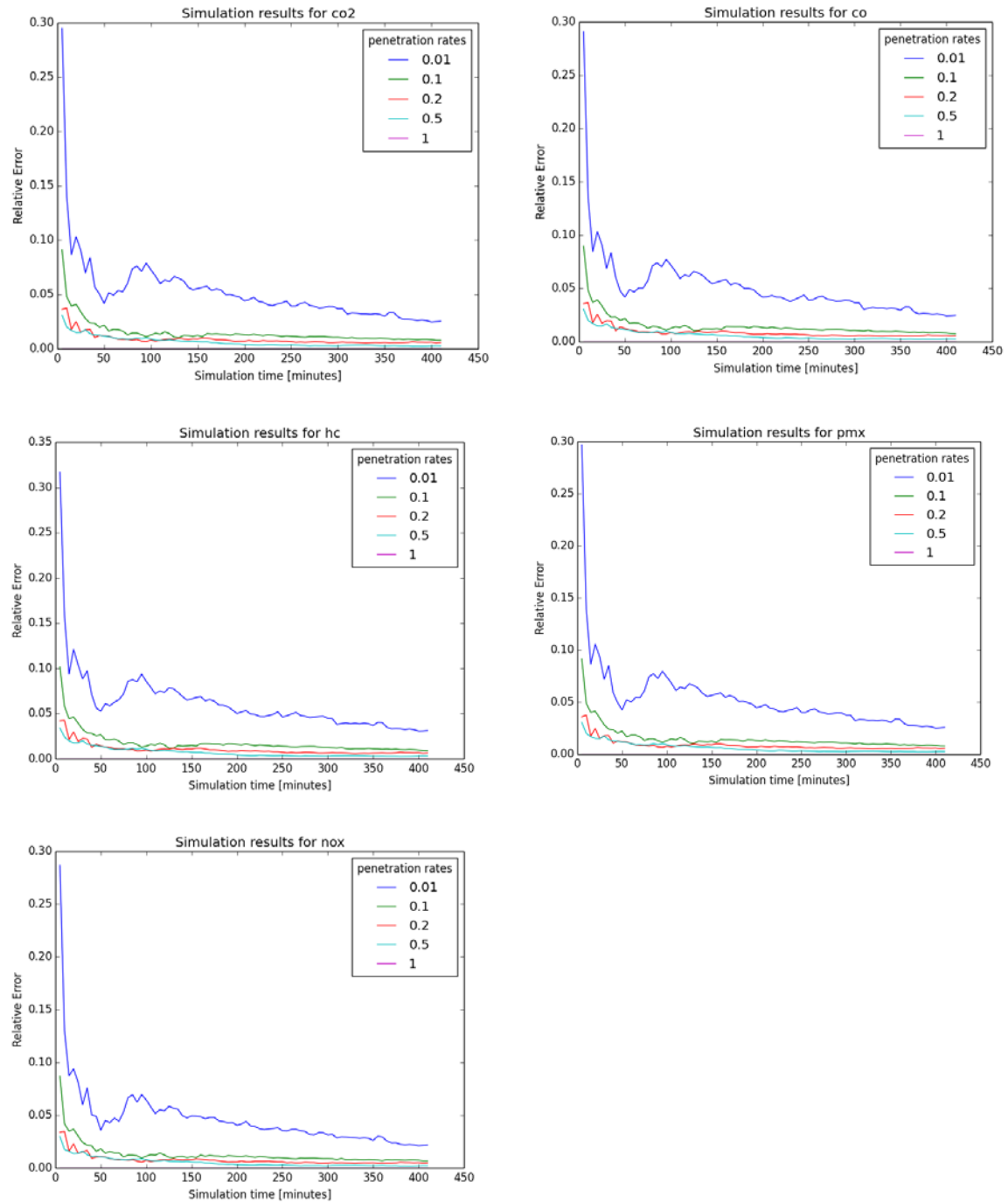


Figure 5.2: Error rates for the first local emissions approach.

6 Cost Analysis

COLOMBO aims at providing a technical alternative to stationary detectors to monitor traffic, using V2X technology even at low penetration rates. Beside the technical feasibility, a major impact factor is the cost difference between these two detecting approaches. The following cost elements need to be taken into account according to, e.g., [eIMPACT, 2006]⁴:

- Investment (Initial costs)
 - System Production and Vehicle Implementation Costs
 - Infrastructure Equipment and Adaption Costs
- Operating and Maintenance Costs (Vehicle and Infrastructure)

Assuming no differences in data and information quality, no improvements in traffic light control and hence in traffic itself are regarded. That means no economic benefits are arising. Thus, this section provides only input numbers supporting a meaningful financial stakeholder analysis, e.g., a Cost-Benefit-Analysis (CBA) or a Return of Investment (RoI), but not the analysis itself. Costs are therefore not broken down into annual slices.

The costs of hardware vary per detector type. The market prices of various common detection technologies are summarized in Table 6.1. Market prices are typically higher than factor costs in an economic CBA, since they include the company's margin, the value added tax (VAT), and subsidies.

Most detectors can cover multiple lanes at once and therefore the costs are summed for 3 lanes, which is an average width for an arm of an intersection. However, some detectors can cover up to 5 lanes and have a cost advantage for larger intersections, while others like the inductive loop and the magnetometer cover only one lane at a time and have an advantage on small intersections. Also note that the detector costs comprise the sensors, additional hardware like video processing for cameras, and the access points. Annual maintenance includes maintenance costs (like cleaning the lenses of the camera) and operation like power consumption.

Table 6.1: Overview of market prices for various detectors and C2X equipment.

Detector Type	Investment cost	Installation costs	Annual maintenance	Lifetime (years)
Inductive loop	€510	€2,500	€10	8
Magnetometer	€1,800	€1,000	€10	10
Weigh-in-motion	€4,000	€2,500	€10	10
Infrared	€2,000	€3,000	€20	10
Acoustic	€8,000	€3,000	€10	8
Radar	€3,000	€3,000	€50	10
Video camera	€1,500	€3,000	€20	10
RSU	€1,000.. €5,000	cf. Table 6.2	€400	15
OBU	€300		€20	10

The road-side unit (RSU) includes the communication interface and the interface to the controller. Neither costs for the controller itself, nor the other parts which are necessary for an unconnected traffic light were considered. Engineering work for signal plans adaptations is also included. Industrial grade RSUs are expected to go down towards €300-€500 when more mass produced components are available compliant to the 802.11(2012) standard. The vehicle investment costs comprise the on-board-unit (OBU) with the communication interface and HMI interface, the maintenance costs are for map updates. Energy consumption is neglected.

⁴ Section 3.5, p.91

For installation of wired connections a key factor is the distance of the sensor to the central data collector. Common activities are listed in Table 6.2. The magnetometers have an advantage over the others regarding installation, because the sensors transmit their measurements wirelessly to an access point at a distance of up to 120 meters.

Table 6.2: Overview of cable installation costs.

Activity	Digging	Digging under pavement	Underground cables	Tube under road surface
Cost per meter	€15	€25	€8	€120

As an example to use the table consider a situation where 3 loops have to be installed at the opposite side of an intersection near the stop line. Suppose the cabinet and the stop line have a distance of 5 meters to the intersection area and the total width of the intersection area is 10 meters square. In that case two trenches of 5 meters are needed and a tube of 10 meters. For connecting the loops 3 underground cables of 20 meters have to be put through the trenches and the underground tube. This results in a total installation cost of €1830.

This example also applies to most other sensors: cameras, radar, acoustic, and infra-red as well as RSUs mounted on a pole which has to be connected to the controller through the same methods as inductive loops, but suffer from extra costs due to safety regulations related to working at the top of a pole. In Table 6.1 it was assumed for the installation costs that an intersection was already in place, so no costs for tubes under the road surface arise, though costs for road closure, equipment rent, permits, digging and new cabling. When installation would be combined with the installation of the traffic light infrastructure, prices would go down by around 50 %.

Sensors that do not require separate power supply cables with voltages in excess of 48 V are the cheapest to install. Most contemporary sensors fall in this category thanks to technologies like Power over Ethernet (PoE). Higher voltages require special shielding to prevent electrocution accidents making both the poles and the cables more expensive. Another aspect to consider is that existing infrastructure near a traffic light can often be re-used. For instance the tubes under the road surface are often already present to connect the signal heads to the controller cabinet. In that case the underground cables for the sensor can simply be pulled through these tubes as well. The same holds for poles to mount sensors or RSUs, street lighting or signal head poles can be reused. The RSU, however, has a clear advantage over sensors since the location is not as much constrained. In an open field a pole can be placed directly next to the controller cabinet for instance, which reduces the installation costs to approximately €1,000 per intersection. In dense urban areas a pole from a signal head can be re-used, but increasing the costs of installation to €3,000, which is the same procedure as for sensors that are placed over the road, like cameras.

The cheapest option of conventional detection, the inductive loop, would cost roughly €12,000 for an entire intersection of 4 arms with annual costs of nearly €1,700. For an RSU this could be estimated at about half (€5,500 initial costs), yielding in €347 annual costs.

7 Summary

For tactical purposes, such as adaptation of traffic lights to changes in traffic demand, traffic management needs on-line information about the state of the managed roads. Nowadays, such applications are realised using stationary detectors that are connected to a traffic light.

As described in Section 2, cooperative traffic surveillance systems available on the market aim rather at offering traffic surveillance to a larger population of individuals. This development ignores the needs of traffic management, although traffic management is the primary actor to mitigate traffic congestions. COLOMBO's traffic surveillance system aims at bringing another evolution into the market, offering solutions from traffic management and road infrastructure providers.

Therefore, the employment of traffic surveillance systems based on vehicular communication was under evaluation within COLOMBO. The major motivation is to offer traffic management cheap and extendible solutions. Four different solutions for collecting information about the state of monitored roads were developed in COLOMBO's WP1 and were presented in Chapter 4 of this document.

The first presented solution is built upon clustering vehicles for compressing the description about the traffic state. It is capable to determine traffic efficiency performance indicators without involving a road side unit. For both reasons, it is interesting to be applied for monitoring larger parts of the road network with a sparse RSU coverage. The second algorithm attempts to determine the density of vehicles determined from the message propagation speed. This is insofar interesting as it may be used on top of all n-hop communication protocols. The implementation in a communication simulator and the subsequent examination shows that the described analytical model is correct and the original idea can be followed in subsequent research. The third approach uses only data from CAMs, messages which are sent by V2X-equipped vehicles periodically. Thereby, it is ready to be deployed as a further "day one" application. The approach is capable to gather a lot of different performance indicators, among them the ones used by the traffic light system developed in COLOMBO's WP2. The fourth algorithm builds upon the traffic flow's dependency on traffic light control. The measure need to operate is the travel time. Albeit the approaches presented herein target on determining this measure via vehicular communication, other (wired and wireless) vehicle re-identification methods could be employed for this purpose as well.

Summarizing, it should be stated that the proposed solutions are not capable to cope with conventional, stationary detectors, yet, if a small amount of equipped vehicles is assumed. It is therefore necessary to:

- a) develop according actor solutions (traffic light algorithms) that are capable to deal with information errors and missing data;
- b) include additional sources of information and join them with the available fine-grained data collected from vehicular communication.

The possibility to obtain meaningful traffic management applications using infrequent and error-prone data only is impressively shown by the fourth solution presented herein and as well targeted in COLOMBO's WP2 (see [COLOMBO D2.3, 2014]). The inclusion of other information sources will be targeted in the subsequent steps of COLOMBO's WP1. Both, other communication modes as well as other participants – pedestrians and bicyclists besides vehicles used so far – will be considered, here.

It should be pointed out that some of the developed solutions could be already easy deployed at existing RSUs. Of course, the according RSU software functionalities had to be extended what raises costs. The outlined benefits may not seem to compensate these costs, but the collected data show a large number of possible applications. They not only feed the third of the solutions presented herein. They are as well one of the key inputs for the subsequent driving anomalies

detection and local emissions monitoring approaches that were outlined in Chapter 5. Further, more sophisticated algorithms could be built upon these data.

When assuming an increase in the number of equipped vehicles, the developed solutions are cheaper than conventional, stationary sensors.

Appendix A – References

- [Akhtar et al., 2012] Akhtar, N; Ergen, S-C; Ozkasap, O, Analysis of Distributed Algorithms for Density Estimation in VANETs, in proc. of the IEEE Vehicular Networking Conference (VNC), 2012.
- [Ancona et al., 2014] Silvia Ancona, Rasvan Stanica, Marco Fiore, “Performance Boundaries of Massive Floating Car Data Offloading”, in proc. of IEEE WONS 2014.
- [Bazzi et al, 2012] A. Bazzi, B.M. Masini, G. Pasolini, “V2V and V2R for Cellular Resources Saving in Vehicular Applications”, IEEE WCNC, Paris, France, Apr.2012.
- [Bellavista, Foschini, Zagmani, 2014] P. Bellavista, L. Foschini, E. Zamagni, "V2X Protocols for Low-Penetration-Rate and Cooperative Traffic Estimations", Proceedings of the 80th IEEE Vehicular Technology Conference-Fall (VTC2014-Fall), pages 1-6, Vancouver, Canada, 14-17 September 2014, IEEE Press.
- [Bellavista, Caselli, Foschini, 2014] P. Bellavista, F. Caselli, L. Foschini, "Implementing and evaluating V2X protocols over iTETRIS: traffic estimation in the COLOMBO project", Proceedings of the 4th ACM Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet'14), held in conjunction with the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'14), pages 1-8, Montreal, Canada, 21-26 September, 2014, ACM Press.
- [Bronsted, Kristensen, 2006] Bronsted, J.; Kristensen, L.M., "Specification and performance evaluation of two zone dissemination protocols for vehicular ad-hoc networks," *Simulation Symposium, 2006. 39th Annual*, vol., no., pp.12 pp., 2-6 April 2006
- [BMVBS, 2012] Federal Ministry of Transport, Building and Urban Development (publisher): ITS Action Plan for the Roads - A framework for the coordinated evolution of existing and the accelerated introduction of new Intelligent Transport Systems in Germany over the period to 2020, 2012, Berlin.
- [COLOMBO D1.1, 2014] COLOMBO project consortium: Scenario Specifications and Required Modifications to Simulation Tools, deliverable 1.1, version updated after the Annual Review, February 2014.
- [COLOMBO D4.3, 2014] COLOMBO project consortium: Pollutant Emission Models and Optimisation, deliverable 4.3, November 2014.
- [COLOMBO D5.1, 2014] COLOMBO project consortium: Pollutant Emission Models and Optimisation, deliverable 4.3, version updated after the Annual Review, February 2014.
- [Daamen et al., 2014] Winnie Daamen, Christine Buisson, Serge P. Hoogendoorn (Editors) (2014) Traffic Simulation and Data: Validation Methods and Applications, CRC Press, ISBN: 978-1482228700
- [DECELL] DECELL (IBM): Cellular Floating Car Data (CFCD) Technology, <http://www.decell.com/Solutions/Technology/fcd.html>
- [Dombush, Joshi, 2007] Dornbush, S.; Joshi, A., "StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's," *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, vol., no., pp.11,15, 22-25 April 2007
- [EC2008/671/EC] European Communication (EU) (2008) EU Decision of 5 August 2008 on the harmonized use of radio spectrum in the 5 875-5 905 MHz frequency band for safety-related applications of Intelligent Transport Systems (ITS)”. Decition 2008/671/EC, EC

- [ECC/DEC/(08)01] Electronic Communications Committee (ECC) (2008) ECC Decision (08)/01 on the harmonized use of the 5875-5925 MHz frequency band for Intelligent Transport Systems (ITS). Decision ECC/DEC/(08)01, ECC.
- [eIMPACT, 2006] eIMPACT project consortium: Methodological framework and database for socio-economic evaluation of Intelligent Vehicle Safety Systems, Deliverable D3, December 2006, available at http://www.eimpact.eu/download/eIMPACT_Deliverable_D3_V10_061214.pdf
- [ES202663, 2009] European Telecommunications Standards Institute (2009) Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band. ES 202 663 V1.1.0, ETSI
- [ETSI-CAM, 2013] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, ETSI EN 302 637-2, v1.3.0, August 2013.
- [ETSI-DENM, 2013] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service, ETSI EN 302 637-3, v1.2.0, August 2013.
- [Fontaine, Smith, 2005] Fontaine, M., Smith, B.: Probe-based Traffic Monitoring Systems with Wireless Location Technology. *Transportation Research Record*(1925), 3-11 (2005).
- [Fontaine, Smith, 2007] Fontaine, M., Smith, B.: Investigation of the Performance of Wireless Location Technology-Based Traffic Monitoring Systems. *Journal of Transportation Engineering*(133), 157-165 (2007).
- [Gehlen et al., 2007] Gehlen, G., Ramme, F., Sories, S., Jodlauk, G.: Cooperative Cars – Using Cellular Communications for Cooperative Automotive Applications. In : 14th ITS World Congress (2007)
- [Geisler et al.] Sandra Geisler, Yuan Chen, Christoph Quix, Guido G. Gehlen, Accuracy Assessment for Traffic Information Derived from Floating Phone Data.
- [Hoogendorn et al., 2011] Hoogendorn, S., Westermann, M., and Hoogendoorn-Lanser, S.: Future scenarios for Traffic Information and Management, TRB 2011 Annual Meeting, Washington D.C.
- [Huber et al.] Werner Huber, Michael Lädke , Rainer Ogger , Extended Floating-Car Data for the Acquisition of Traffic Information, White Paper.
- [Ibrahim, Weigle, 2008] Ibrahim, K.; Weigle, M.C., "Optimizing CASCADE data aggregation for VANETs," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on* , vol., no., pp.724,729, Sept. 29 2008-Oct. 2 2008
- [IEEE 802.11p-2010] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Wireless Access in Vehicular Environments, 2010.
- [IEEE 802.11-2012] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.
- [Jerbi et al., 2007] Jerbi, M.; Senouci, S.-M.; Rasheed, T.; Ghamri-Doudane, Y., "An Infrastructure-Free Traffic Information System for Vehicular Networks," *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th* , vol., no., pp.2086,2090, Sept. 30 2007-Oct. 3 2007

- [Jost et al, 2011] Jost, H., S. Köhler, F. Köster (2011) Towards a Safer Development of Driver Assistance Systems by Applying Requirements-Based Methods. 14th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2011) , 5.-7. Okt. 2011, Washington D.C., USA.
- [Karim, Adeli, 2002] Karim, A. and H. Adeli, Incident Detection Algorithm using Wavelet Energy Representation of Traffic Patterns. J. Transp. Eng., Vol. 128, No. 3, 2002, p. 232Ð242.
- [Lochert et al., 2008] Lochert, C.; Scheuermann, B; Wewetzer, C; Luebke, A; Mauve, M, Data Aggregation and Roadside Unit Placement for a Vanet Traffic Information System, in Proc. of the Fifth ACM International Workshop on VehiculAr Inter-NETworking, pp. 58—65, 2008 San Francisco, California, USA, 2008
- [Meysel, 2008] Meysel, F., Map-based Multi-Object Dynamic Scene Description and Atypical Event Detection. Ph.D. thesis, Humboldt Universität zu Berlin, 2008.
- [MobileMilenium] The Mobile Milenium Project: <http://traffic.berkeley.edu/>
- [Modsching et al., 2006] Marko Modsching, Ronny Kramer, Klaus ten Hagen, “Field trial on GPS Accuracy in a medium size city: The influence of built-up”, in proc. of the 3rd Workshop on Positioning, Navigation and Communication, 2006.
- [Nadeem et al., 2004] Nadeem, T.; D. Sasan, Liao, C; Iftode, L., TrafficView: Traffic Data Dissemination using Car-to-Car Communication , *ACM Mobile Computing and Communications Review (MC2R), Special Issue on Mobile Data Management*, Vol. 8, No. 3, July 2004, pp. 6-19
- [Neumann, Wagner, 2012] Neumann, T.; Wagner, P. (2012) DPAnalyzer - Rückstaulängenschätzung mit Floating-Car-Daten für das Qualitätsmanagement an Lichtsignalanlagen. Straßenverkehrstechnik, 56 (6), Seiten 353-360. Kirschbaum Verlag
- [Saul, Kozempel, Haberjahn, 2014] Saul, H., K. Kozempel, M. Haberjahn, A Comparison of Methods for Detecting atypical Trajectories, presented in: Urban Transport XX, 138, Seiten 393-403. WITPress. 20th International Conference on Urban Transport and the Environment (2014)
- [SODIT] SODIT - Real-time information on road traffic based on Floating Car Data, Laurent Breheret, ANR France
- [Stanica et al., 2013] R. Stanica, M. Fiore, F. Malandrino, “Offloading Floating Car Data”, IEEE WoWMoM, Madrid, Spain Jun.2013
- [Tang, Gao, 2005] Tang, S. and H. Gao, Traffic-incident detection-algorithm based on nonparametric regression. Intelligent Transportation Systems, IEEE Transactions on, Vol. 6, No. 1, 2005, pp. 38–42.
- [TomTom] White paper How TomTom’s HD Traffic™ and IQ Routes™ data provides the very best routing.
- [Wegener et al., 2007] Wegener, A. ; Hellbruck, H. ; Fischer, S. ; Schmidt, C. ; Fekete, S., AutoCast: An Adaptive Data Dissemination Protocol for Traffic Information Systems, *VTC '07-Spring: Proceedings of the 66th IEEE Vehicular Technology Conference*, pp. 1947-1951
- [Weil et al., 1998] Weil, R., J.Wootton, and A. GarcŠa-Ortiz, Traffic incident detection: Sensors and algorithms. Mathematical and Computer Modelling, Vol. 27, No. 9Ð11, 1998, pp. 257 – 291.
- [Wischhof et al, 2003a] Wischhof, L.; Ebner, A.; Rohling, H.; Lott, M.; Halfmann, R., "Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE* , vol., no., pp.6,11, 9-11 June 2003

- [Wischhof et al., 2003b] Wischhof, L.; Ebner, A.; Rohling, H.; Lott, M.; Halfmann, R, SOTIS - A Self-Organizing Traffic Information System, VTC '03-Spring: Proceedings of the 57th IEEE Vehicular Technology Conference
- [Wischhof et al., 2005] Wischhof, L.; Ebner, A.; Rohling, H., "Information dissemination in self-organizing intervehicle networks," *Intelligent Transportation Systems, IEEE Transactions on* , vol.6, no.1, pp.90,101, March 2005
- [Xu, Barth, 2006a] Huaying Xu, Matthew Barth, Travel Time Estimation Techniques for Traffic Information Systems Based on Intervehicle Communications, Transportation Research Record: Journal of the Transportation Research Board, vol. 1944, 2006.
- [Xu, Barth, 2006] Huaying Xu; Barth, M., "An adaptive dissemination mechanism for inter-vehicle communication-based decentralized traffic information systems," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE* , vol., no., pp.1207,1213, 17-20 Sept. 2006

Appendix B – Further iTETRIS Platform Extensions

Beside the iTETRIS extensions described in D5.1, the integrations of the traffic surveillance algorithms described in Section 4 required further extensions, which are described in the following sections.

B.1 Application Module Extensions

As previously mentioned the iCS application developed introduces an adapter module between the platform and the implementation of the communication protocols, with the main goal of increasing the level of abstraction, separating all the iCS-specific interactions, and offering a simpler interface for traditional simulation developers. The lower level realizes several basic functions for the interaction with the iCS framework while offering to the protocol level an abstraction to easily send and receive messages. The main components of the application are listed below.

The *connection server* is a component that manages all the communications with iCS and interprets the messages to and from the framework, converting them to the representation used by the rest of the application. Furthermore, this component informs the rest of the application of the beginning of a new simulation step and it commands the removal of nodes to *node handler* when necessary.

The *node handler* contains a list of all the nodes currently active in the simulation. This module sits between the *connection server* and the *nodes*: every time iCS sends a command to the application, the server decodes the message and then it calls the appropriate functionality in the *node handler* who selects the correct node and forwards to it the request. This component also manages the creation of new nodes and their removal.

The *iCS helper* module realises a series of support functionality such as a component that contains a series of primitives to obtain the appropriate subscription message to send to the iCS framework, hiding the specific formats the other components of the application.

The *node* is one of the most important component of the applications, because it manages the subscriptions and offers an interface that allows the interaction between the two levels of the application. This module offers the usual communication primitives to the protocol, hiding all the interaction through subscriptions with the iCS framework. It also makes available to the protocol the position and the speed of the node, gathered from sumo through the appropriate iCS subscription.

The *protocol helper* module realizes several support functionality used by the implementation of the protocol, such as a payload storage that allows the application to send only the identifier of a payload maintained to the framework, thus making more efficient and lightweight the messages exchanged with iCS. This module also realizes a scheduler that allows the execution of a specified action at a future time, a functionality that is frequently used by the logic of the protocol, and implements the same random generator used by ns-3.

The *ITS controller* realizes common functionality used by the different logics of the protocol. It manages their setup and configuration, the activation and deactivation of the appropriate logic when the nodes enters or exit from a group, and it filters the incoming messages according to their destination and the particular message type.

The *sampler* is a component tasked with the introduction of random errors to the position and the speed of a node to better simulate the imprecisions of the real sensors. This modules samples the values from iCS at fixed intervals and applies the appropriate errors to them. The magnitude of the errors introduced by this module can be specified in the configuration file, to adapt it to different classes of sensors.

The *ITS protocols* are a series of components that model a specific functionality of the protocol, such as the member of a group, its leader or the RSU logic.

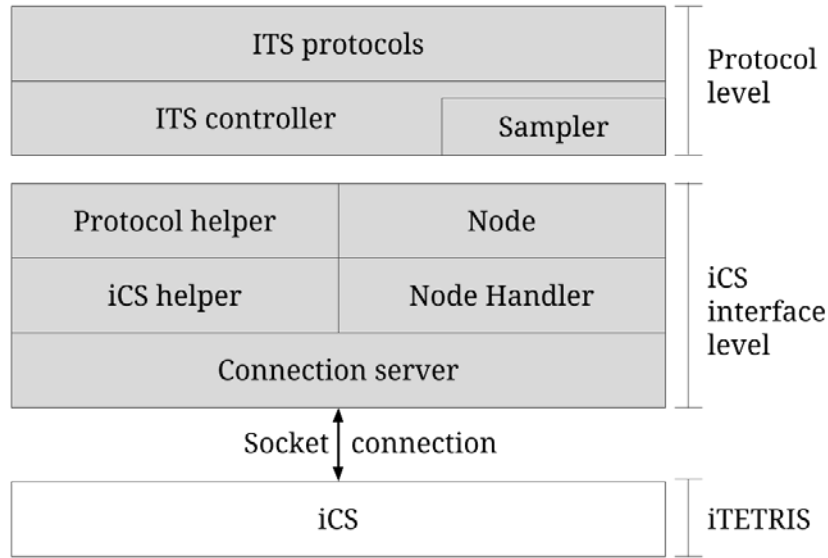


Figure B.1: Architecture of the enhanced application module of iTETRIS.

B.2 iCS optimization

While porting their V2V-based traffic surveillance from ns-3 to the iCS, UNIBO spent much effort on improving the platform. On the one hand, UNIBO worked on the iCS-based application and developed a middleware layer that works as a stub simplifying network programming aspects: it introduces the abstraction of Node and, for each Node, provides send/receive primitives hiding the complexity of subscriptions used by iCS to control communication and exchange data. On the other hand, UNIBO worked on removing some bugs and making the simulation execution more efficient. This improvement task was quite time consuming, but increased the quality of the simulation platform relevantly in terms of both obtained simulation results (which are now aligned with those obtainable with standalone ns3) and time for completing a simulation (which dropped significantly). One of the main technical challenges UNIBO had to overcome was the difference in handling event scheduling in the ns-3 simulator alone and in the integrated iTETRIS platform. In fact, the coordination between the various parts of the iCS architecture occurs at fixed intervals, so the events lose the granularity possible in a discrete event simulator like ns-3. To partially mitigate this problem, the frequency of the coordination instants has been reduced from the default value of 1 s to a minimum value of 1 ms (in the reported simulations they are in the 10-25 ms range). Another non-negligible issue was the occurrence of a large number of collisions caused in iTETRIS by the scheduling of multiple messages at the same instant at the beginning of a coordination step, and not present in more only-network-oriented simulation tools such as ns-3. To avoid this problem, we have worked on evolving the iTETRIS framework in order to include the ability to schedule events at arbitrary intervals, as in ns-3, thus avoiding most of the previously experienced collisions.

As shown in Figure B.2, the performance results collected for our protocol implementation as an iCS application is now comparable with the alternate ns-3-only implementation, while the less optimized versions provided worse results.

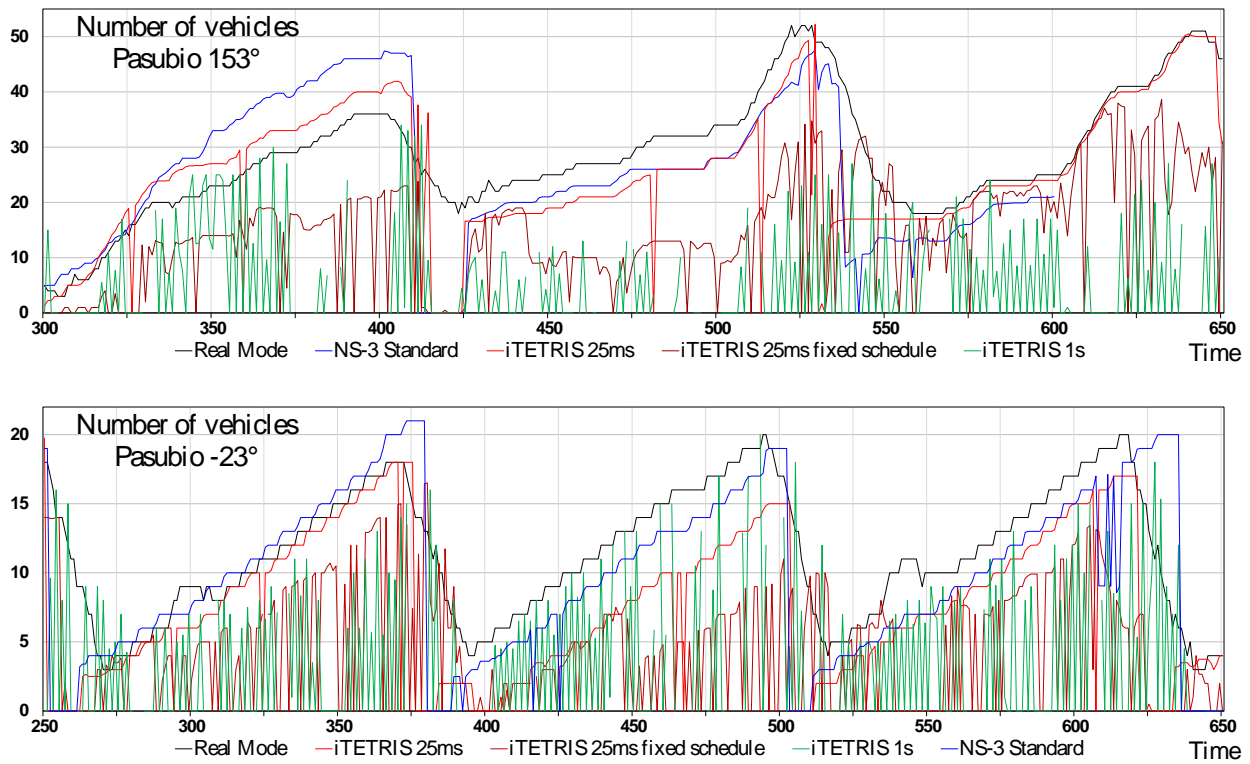


Figure B.2: Evolution of the iCS implementation: the change in event scheduling has a great impact on the performances.